

IBM[®] DB2[®] Spatial Extender e
Geodetic Extender



Guida di riferimento per l'utente

Versione 8.2

IBM[®] DB2[®] Spatial Extender e
Geodetic Extender



Guida di riferimento per l'utente

Versione 8.2

Prima di utilizzare queste informazioni e il prodotto supportato, accertarsi di aver consultato le informazioni generiche contenute in *Informazioni particolari*.

Questo documento contiene informazioni di proprietà dell'IBM. Viene fornito con un accordo di licenza ed è protetto dalle leggi sul copyright. Le informazioni contenute in questa pubblicazione non includono alcuna garanzia sul prodotto e tutte le istruzioni fornite in questo manuale non vanno interpretate in tale senso.

E' possibile ordinare le pubblicazioni IBM in linea oppure tramite un rappresentante IBM.

- Per ordinare le pubblicazioni in linea, visitare IBM Publications Center all'indirizzo www.ibm.com/shop/publications/order
- Per contattare un rivenditore IBM locale, visitare la pagina IBM Directory of Worldwide Contacts all'indirizzo www.ibm.com/planetwide

Per ordinare le pubblicazioni DB2 da DB2 Marketing and Sales negli Stati Uniti o Canada, chiamare il numero 1-800-IBM-4YOU (426-4968).

Quando si inviano informazioni all'IBM, si garantisce il diritto non esclusivo all'IBM di utilizzo o distribuzione di queste informazioni nel modo ritenuto più opportuno senza alcun obbligo nei confronti dell'utente.

© Copyright International Business Machines Corporation 1998, 2004. Tutti i diritti riservati.

Indice

Parte 1. Introduzione 1

Capitolo 1. Informazioni su DB2 Spatial Extender 3

Uso di DB2 Spatial Extender	3
Dati spaziali e geodetici	4
Modalità di rappresentazione delle caratteristiche geografiche nei dati	4
Struttura dei dati spaziali	5
Natura dei dati geodetici	6
Provenienza dei dati spaziali	6
Interazione tra funzioni, informazioni spaziali, dati spaziali e forme geometriche	8

Capitolo 2. Informazioni sulle forme geometriche 11

Forme geometriche	11
Proprietà delle forme geometriche	13
Tipo	13
Coordinate della forma geometrica	13
Coordinate X e Y	14
Coordinate Z	14
Coordinate M.	14
Spazio interno, contorno e spazio esterno	14
Semplice o non semplice	14
Chiusa	14
Vuote o non vuote	14
MBR (minimum bounding rectangle).	14
Dimensione	15
Identificativo del sistema di riferimento spaziale	15

Capitolo 3. Uso di DB2 Spatial Extender 17

Uso di DB2 Spatial Extender.	17
Interfacce di DB2 Spatial Extender e funzionalità associate	17
Impostazione di DB2 Spatial Extender e creazione di progetti	17

Parte 2. Impostazione di DB2 Spatial Extender 23

Capitolo 4. Introduzione a DB2 Spatial Extender 25

Impostazione e installazione di Spatial Extender	25
Impostazione e installazione di Spatial Extender	25
Requisiti di sistema per l'installazione di Spatial Extender	26
Installazione di DB2 Spatial Extender per Windows	27
Installazione di DB2 Spatial Extender per AIX.	29
Installazione di DB2 Spatial Extender per HP-UX	31

Installazione di DB2 Spatial Extender per Solaris Operating Environment	33
Installazione di DB2 Spatial Extender per Linux	35
Creazione dell'istanza di DB2 Spatial Extender	37
Verifica dell'installazione di Spatial Extender	39
Risoluzione dei problemi di installazione	40
Considerazioni successive all'installazione	41
Download di ArcExplorer per DB2	41
Accesso ai dati di riferimento del geocoder.	41
CD per dati e carte geografiche di DB2 Spatial Extender	42

Capitolo 5. Migrazione dell'ambiente Spatial Extender a DB2 Universal Database Versione 8 45

Migrazione di un database abilitato per le funzioni spaziali.	45
Messaggi di migrazione	46
Comando db2se migrate_v82	47

Capitolo 6. Impostazione di un database 49

Configurazione di un database per l'inserimento dei dati spaziali	49
Ottimizzazione dei parametri di configurazione del database	49
Ottimizzazione delle caratteristiche della registrazione della transazione	49
Ottimizzazione della dimensione heap dell'applicazione.	51
Ottimizzazione della dimensione heap di controllo dell'applicazione	51

Capitolo 7. Impostazione delle risorse spaziali per un database 53

Impostazione delle risorse nel database	53
Inventario delle risorse per il database	53
Abilitazione di un database per le operazioni spaziali.	54
Uso dei dati di riferimento	54
Dati di riferimento	55
Impostazione dell'accesso ai dati di riferimento	55
Registrazione di un geocoder	56

Parte 3. Creazione di progetti che utilizzano dati spaziali 57

Capitolo 8. Impostazione delle risorse spaziali per un progetto 59

Uso dei sistemi di coordinate	59
Sistemi di coordinate	59
Sistema di coordinate geografiche	59
Sistemi di coordinate proiettate.	65

Indice

Selezione o creazione dei sistemi di coordinate	66
Impostazione dei sistemi di riferimento spaziali	68
Sistemi di riferimento spaziali	68
Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo	69
Sistemi di riferimento spaziali forniti con DB2 Spatial Extender	71
Fattori di conversione per la trasformazione delle coordinate in numeri interi	73
Creazione di un sistema di riferimento spaziale	75
Calcolo dei fattori di scala	77
Determinazione delle coordinate e misure minime e massime	78
Calcolo dei valori di offset	79

Capitolo 9. Impostazione delle colonne spaziali 81

Colonne spaziali	81
Colonne spaziali con contenuto visualizzabile	81
Tipi di dati spaziali	81
Creazione di colonne spaziali	83
Registrazione delle colonne spaziali	85

Capitolo 10. Inserimento dei dati nelle colonne spaziali. 87

Importazione ed esportazione dei dati spaziali	87
Informazioni sull'importazione e l'esportazione dei dati spaziali	87
Importazione di dati spaziali	88
Esportazione dei dati spaziali	91
Uso del geocoder	92
Geocoder e funzioni di geocoding	93
Impostazione delle funzioni di geocoding	94
Impostazione dell'esecuzione automatica del geocoder	96
Esecuzione del geocoder in modalità batch	97

Capitolo 11. Uso degli indici e delle viste per accedere ai dati spaziali . . . 99

Tipi di indici spaziali	99
Indici di griglia spaziali	100
Generazione degli indici di griglia spaziale	100
Uso delle funzioni spaziali in un'interrogazione	101
Come un'interrogazione utilizza gli indici di griglia spaziali	101
Considerazioni sul numero di livelli di indice e dimensioni della griglia	102
Numero dei livelli di griglia	102
Dimensione delle celle della griglia	103
Creazione degli indici di griglia spaziali	106
Istruzione CREATE INDEX per l'indice di griglia spaziale	108
Regolazione degli indici di griglia spaziali con Index Advisor	110
Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica	110
Determinazione delle dimensioni della griglia per un indice di griglia spaziale	111

Analisi delle statistiche dell'indice di griglia spaziale	112
Comando gseidx	117
Utilizzo delle viste per accedere alle colonne spaziali	120

Capitolo 12. Analisi e generazione delle informazioni spaziali 121

Ambienti di esecuzione delle analisi spaziali	121
Esempi delle operazioni eseguite dalle funzioni spaziali	121
Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni	122

Capitolo 13. Comandi di DB2 Spatial Extender 125

Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti	125
---	-----

Capitolo 14. Scrittura di applicazioni ed utilizzo del programma di esempio. 133

Scrittura di applicazioni per DB2 Spatial Extender	133
Inserimento del file di intestazione di DB2 Spatial Extender nelle applicazioni spaziali	133
Richiamo delle procedure memorizzate di DB2 Spatial Extender da un'applicazione	134
Programma di esempio DB2 Spatial Extender	135

Capitolo 15. Individuazione dei problemi con DB2 Spatial Extender . . 143

Interpretazione dei messaggi di DB2 Spatial Extender	143
Parametri di output delle procedure memorizzate di DB2 Spatial Extender	145
Messaggi delle funzioni di DB2 Spatial Extender	147
Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender	149
Messaggi del Centro di controllo DB2	151
Registrazione dei problemi di DB2 Spatial Extender mediante il comando db2trc	152
File di notifica per la gestione	153

Parte 4. Utilizzo di DB2 Geodetic Extender 155

Capitolo 16. DB2 Geodetic Extender 157

DB2 Geodetic Extender	157
Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender	158
Datum geodetici	158
Latitudine e longitudine geodetiche	159
Distanze geodesiche	160
Regioni geodetiche	161

Capitolo 17. Impostazione di DB2 Geodetic Extender 165

Impostazione ed abilitazione di DB2 Geodetic Extender	165
---	-----

Migrazione da Informix Geodetic DataBlade a DB2
 Geodetic Extender 166
 Inserimento dei dati geodetici nelle colonne
 spaziali 173

Capitolo 18. Indici geodetici 175

Indici geodetici Voronoi 175
 Strutture delle celle Voronoi 176
 Considerazioni per la selezione di una struttura di
 celle Voronoi alternativa 177
 Creazione di indici geodetici Voronoi 179
 Istruzione CREATE INDEX per l'indice geodetico
 Voronoi 180
 Strutture delle celle Voronoi fornite con DB2
 Geodetic Extender 182
 Mondiale, basata sulla densità di popolazione
 (ID Voronoi: 1) 183
 Stati Uniti (ID Voronoi: 2) 184
 Canada (ID Voronoi: 3) 185
 India (ID Voronoi: 4) 186
 Giappone (ID Voronoi: 5) 187
 Africa (ID Voronoi: 6) 188
 Australia (ID Voronoi: 7) 189
 Europa (ID Voronoi: 8) 190
 America del Nord (ID Voronoi: 9) 191
 America del Sud (ID Voronoi: 10) 192
 Mediterraneo (ID Voronoi: 11) 193
 Mondiale, distribuzione dati uniforme,
 risoluzione media – dodeca04 (ID Voronoi: 12) 194
 Mondiale, nazioni industriali – nazioni G7 (Id
 Voronoi: 13) 195
 Mondiale, distribuzione dati uniforme,
 risoluzione bassa – isotipo (ID Voronoi: 14) 196

**Capitolo 19. Differenze nell'utilizzo dei
 dati geodetici e spaziali 197**

Attributi x e y minimi e massimi 197
 Differenze nell'uso di rappresentazioni della Terra
 piana o sferica 197
 Segmenti di linee che attraversano il 180°
 meridiano 198
 Poligoni a cavallo del 180° meridiano 199
 Poligoni che contengono un polo 202
 Poligoni che rappresentano gli emisferi, le
 cinture equatoriali e l'intero pianeta 203
 Funzioni spaziali supportate da DB2 Geodetic
 Extender 206
 Procedure memorizzate e viste del catalogo di DB2
 Geodetic Extender 211
 Datum supportati da DB2 Geodetic Extender 211
 Sferoidi geodetici 220

Parte 5. Materiale di riferimento 223

Capitolo 20. Procedure memorizzate 225

GSE_export_sde 226
 GSE_import_sde 228
 ST_alter_coordsys 230
 ST_alter_srs 232

ST_create_coordsys 236
 ST_create_srs 238
 ST_disable_autogeocoding 244
 ST_disable_db 246
 ST_drop_coordsys 248
 ST_drop_srs 249
 ST_enable_autogeocoding 250
 ST_enable_db 252
 ST_export_shape 254
 ST_import_shape 257
 ST_register_geocoder 265
 ST_register_spatial_column 270
 ST_remove_geocoding_setup 271
 ST_run_geocoding 273
 ST_setup_geocoding 276
 ST_unregister_geocoder 280
 ST_unregister_spatial_column 281

Capitolo 21. Viste del catalogo 283

Vista del catalogo
 DB2GSE.ST_COORDINATE_SYSTEMS 283
 Vista del catalogo
 DB2GSE.ST_GEOMETRY_COLUMNS 284
 Vista del catalogo DB2GSE.ST_GEOCODER_
 PARAMETERS 285
 Vista del catalogo DB2GSE.ST_GEOCODERS 287
 Vista del catalogo DB2GSE.ST_GEOCODING 287
 Vista del catalogo
 DB2GSE.ST_GEOCODING_PARAMETERS 288
 Vista del catalogo DB2GSE.ST_SIZINGS 290
 Vista del catalogo
 DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS 291
 Vista del catalogo
 DB2GSE.ST_UNITS_OF_MEASURE 293

**Capitolo 22. Funzioni spaziali:
 categorie e uso 295**

Funzioni spaziali 295
 Funzioni spaziali che convertono i valori delle
 forme geometriche in formati di scambio dati 295
 Panoramica sulle funzioni di costruzione 296
 Funzioni che operano su formati di scambio dati 296
 Funzione per la creazione di forme geometriche
 da coordinate 297
 Esempi 298
 Conversione in rappresentazioni WKT (well-known
 text) 300
 conversione in rappresentazioni WKB (well-known
 binary) 301
 Conversione in rappresentazioni shape ESRI 302
 Conversione in rappresentazioni GML (Geography
 Markup Language) 303
 Funzioni che confrontano le funzioni geografiche 304
 Panoramica delle funzioni di confronto 304
 Elenco delle funzioni 306
 Funzioni che controllano se una forma geometrica
 ne contiene un'altra 306
 ST_Contains 306
 ST_Within 308

Indice

Funzioni che controllano le intersezioni tra forme geometriche	309
ST_Intersects	310
ST_Crosses	311
ST_Overlaps	312
ST_Touches	313
Funzioni che confrontano gli involuipi delle forme geometriche	315
ST_EnvIntersects	315
ST_MBRIntersects	315
Funzioni che controllano la similitudine di due elementi	315
ST_EqualCoordsys	315
ST_Equals	316
ST_EqualSRS	316
Funzioni che controllano la presenza di intersezioni tra due forme geometriche	317
Funzioni che confrontano forme geometriche alla stringa matrice di modello DE-9IM	318
Funzioni che restituiscono di informazioni sulle proprietà delle forme geometriche	318
Funzioni che restituiscono informazioni sul tipo di dati	318
Funzioni che restituiscono informazioni sulle coordinate e le misure	319
ST_CoordDim	319
ST_IsMeasured	319
ST_IsValid	319
ST_Is3D	319
ST_M	320
ST_MaxM	320
ST_MaxX	320
ST_MaxY	320
ST_MaxZ	320
ST_MinM	320
ST_MinX	320
ST_MinY	320
ST_MinZ	320
ST_X	320
ST_Y	320
ST_Z	320
Funzioni che restituiscono informazioni sulle forme geometriche interne a una forma geometrica	321
ST_Centroid	321
ST_EndPoint	321
ST_GeometryN	321
ST_LineStringN	321
ST_MidPoint	321
ST_NumGeometries	321
ST_NumLineStrings	322
ST_NumPoints	322
ST_NumPolygons	322
ST_PointN	322
ST_PolygonN	322
ST_StartPoint	322
Funzioni che mostrano informazioni su contorni, involuipi e circonferenze	322
ST_Boundary	322
ST_Envelope	323
ST_EnvIntersects	323
ST_ExteriorRing	323
ST_InteriorRingN	323
ST_MBR	323
ST_MBRIntersects	323
ST_NumInteriorRing	323
ST_Perimeter	323
Funzioni che restituiscono informazioni sulle dimensioni di una forma geometrica	323
ST_Area	324
ST_Dimension	324
ST_Length	324
Funzioni che rivelano se una forma geometrica è chiusa, vuota o semplice.	324
ST_IsClosed	324
ST_IsEmpty	324
ST_IsSimple	324
Funzioni che identificano il sistema di riferimento spaziale di una forma geometrica.	324
ST_SrsId (chiamata anche ST_SRID)	324
ST_SrsName	325
Funzioni che generano nuove forme geometriche da forme geometriche esistenti	325
Funzioni che convertono una forma geometrica in un'altra	325
ST_Polygon	325
ST_ToGeomColl	325
ST_ToLineString	326
ST_ToMultiLine	326
ST_ToMultiPoint	326
ST_ToMultiPolygon	326
ST_ToPoint	326
ST_ToPolygon	326
Funzioni che creano nuove forme geometriche con diverse configurazioni di spazio	326
ST_Buffer	326
ST_ConvexHull	327
ST_Difference	328
ST_Intersection	328
ST_SymDifference	329
Funzioni che estrapolano una forma geometrica da molte	330
Aggregazione MBR	330
ST_Union	330
Aggregazione unione	330
Funzioni che estrapolano nuove forme geometriche in base alle misure	330
ST_FindMeasure (chiamata anche ST_LocateAlong)	331
ST_MeasureBetween (chiamata anche ST_LocateBetween)	331
Funzioni che creano forme modificate di forme geometriche esistenti	331
ST_AppendPoint	331
ST_ChangePoint	332
ST_Generalize	332
ST_M	332
ST_PerpPoints	332
ST_RemovePoint	332
ST_X	332
ST_Y	332
ST_Z	332

Funzioni che restituiscono informazioni sulla
distanza 332
Funzioni che restituiscono informazioni sull'indice 333
Conversioni tra sistemi di coordinate 333

**Capitolo 23. Funzioni spaziali: sintassi
e parametri 335**

Funzioni spaziali: considerazioni e tipi di dati
associati 335
 Fattori da considerare 335
 Gestione dei valori di ST_Geometry come valori
di un tipo secondario. 336
 Funzioni spaziali elencate in base al tipo di
input 337
EnvelopesIntersect. 339
Aggregazione MBR 341
ST_AppendPoint 343
ST_Area 344
ST_AsBinary 348
ST_AsGML 349
ST_AsShape 350
ST_AsText 351
ST_Boundary 352
ST_Buffer. 354
ST_Centroid 357
ST_ChangePoint 358
ST_Contains 359
ST_ConvexHull. 361
ST_CoordDim 363
ST_Crosses 364
ST_Difference 365
ST_Dimension 367
ST_Disjoint 368
ST_Distance 370
ST_Edge_GC_USA 373
ST_Endpoint 377
ST_Envelope 378
ST_EnvIntersects 379
ST_EqualCoordsys. 380
ST_Equals 381
ST_EqualSRS 383
ST_ExteriorRing 384
ST_FindMeasure o ST_LocateAlong 385
ST_Generalize 387
ST_GeomCollection 388
ST_GeomCollFromTxt 390
ST_GeomCollFromWKB 392
ST_Geometry 393
ST_GeometryN 395
ST_GeometryType 396
ST_GeomFromText 397
ST_GeomFromWKB 398
ST_GetIndexParms 400
ST_InteriorRingN 402
ST_Intersection 403
ST_Intersects 405
ST_Is3d 407
ST_IsClosed 408
ST_IsEmpty 409
ST_IsMeasured 410
ST_IsRing 411

ST_IsSimple 412
ST_IsValid 414
ST_Length 415
ST_LineFromText 417
ST_LineFromWKB. 418
ST_LineString 419
ST_LineStringN 420
ST_M 422
ST_MaxM 423
ST_MaxX 425
ST_MaxY 426
ST_MaxZ 428
ST_MBR 429
ST_MBRIntersects 430
ST_MeasureBetween, ST_LocateBetween 432
ST_MidPoint 434
ST_MinM. 435
ST_MinX 436
ST_MinY 438
ST_MinZ 439
ST_MLineFromText 440
ST_MLineFromWKB 442
ST_MPointFromText 444
ST_MPointFromWKB. 445
ST_MPolyFromText 446
ST_MPolyFromWKB 448
ST_MultiLineString 450
ST_MultiPoint 451
ST_MultiPolygon 453
ST_NumGeometries 455
ST_NumInteriorRing 456
ST_NumLineStrings 457
ST_NumPoints 458
ST_NumPolygons 459
ST_Overlaps. 460
ST_Perimeter 462
ST_PerpPoints 463
ST_Point 465
ST_PointFromText 468
ST_PointFromWKB 469
ST_PointN 471
ST_PointOnSurface 472
ST_PolyFromText 473
ST_PolyFromWKB. 474
ST_Polygon 475
ST_PolygonN 478
ST_Relate. 479
ST_RemovePoint 480
ST_SrsId, ST_SRID 481
ST_SrsName. 482
ST_StartPoint 483
ST_SymDifference 484
ST_ToGeomColl 487
ST_ToLineString 488
ST_ToMultiLine 489
ST_ToMultiPoint 490
ST_ToMultiPolygon 491
ST_ToPoint 492
ST_ToPolygon 493
ST_Touches 494
ST_Transform 495

Indice

ST_Union	497
ST_Within	499
ST_WKBToSQL	501
ST_WKTTToSQL	502
ST_X	503
ST_Y	504
ST_Z	506
Aggregazione unione	507

Capitolo 24. Gruppi di trasformazione 509

Gruppi di trasformazione	509
Gruppo di trasformazione ST_WellKnownText	509
Gruppo di trasformazione ST_WellKnownBinary	510
Gruppo di trasformazione ST_Shape	512
Gruppo di trasformazione ST_GML	513

Capitolo 25. Formati dati supportati 515

Rappresentazione WKT (well-known text)	515
Rappresentazione WKB (well-known binary)	520
Rappresentazione shape	522
Rappresentazione GML (Geography Markup Language)	522

Capitolo 26. Sistemi di coordinate supportati 525

Sistemi di coordinate supportati	525
Sintassi dei sistemi di coordinate	525
Unità di misura angolari supportate	527
Sferoidi supportati	528
Datum geodetici supportati	529
Primi meridiani supportati	531
Proiezioni di mappa supportate	531

Appendice A. Procedure memorizzate obsolete 535

db2gse.gse_enable_autogc	535
db2gse.gse_enable_db	537
db2gse.gse_enable_idx	538
db2gse.gse_enable_sref	539
db2gse.gse_export_shape	541
db2gse.gse_disable_autogc	542
db2gse.gse_disable_db	544
db2gse.gse_disable_sref	544
db2gse.gse_import_shape	545
db2gse.gse_register_gc	547
db2gse.gse_register_layer	548
db2gse.gse_run_gc	553
db2gse.gse_unregist_gc	554
db2gse.gse_unregist_layer	555

Appendice B. Viste del catalogo obsolete 557

DB2GSE.COORD_REF_SYS	557
DB2GSE.GEOMETRY_COLUMNS	557
DB2GSE.SPATIAL_GEOCODER	558
DB2GSE.SPATIAL_REF_SYS	558

Appendice C. Funzioni spaziali obsolete 561

AsShape	562
GeometryFromShape	562
Is3d	562
IsMeasured	562
LineFromShape	563
LocateAlong	563
LocateBetween	563
M	564
MLine FromShape	564
MPointFromShape	564
MPolyFromShape	564
PointFromShape	565
PolyFromShape	565
ShapeToSQL	565
ST_GeomFromText	566
ST_GeomFromWKB	566
ST_LineFromText	566
ST_LineFromWKB	566
ST_MLineFromText	567
ST_MLineFromWKB	567
ST_MPointFromText	567
ST_MPointFromWKB	568
ST_MPolyFromText	568
ST_MPolyFromWKB	568
ST_OrderingEquals	569
ST_Point	569
ST_PointFromText	569
ST_PolyFromText	569
ST_PolyFromWKB	570
ST_Transform	570
ST_SymmetricDiff	570
Z	571

Informazioni particolari 573

Marchi	575
------------------	-----

Indice analitico 577

Come ottenere ulteriori informazioni dalla IBM 583

Informazioni sul prodotto	583
-------------------------------------	-----

Parte 1. Introduzione

Capitolo 1. Informazioni su DB2 Spatial Extender

Il capitolo introduce DB2 Spatial Extender descrivendone le funzionalità e i dati supportati.

Uso di DB2 Spatial Extender

DB2® Spatial Extender può essere utilizzato per generare e analizzare informazioni di carattere geografico e per memorizzare e gestire i dati che costituiscono questo tipo di informazioni. Per *caratteristica geografica* (a volte chiamata semplicemente *caratteristica* in questa pubblicazione) si intende qualsiasi elemento nel mondo reale che abbia una posizione identificabile, o qualsiasi elemento che possa essere immaginato come effettivamente presente in una ubicazione reale. Una caratteristica geografica potrebbe essere:

- Un oggetto, ovvero un elemento concreto di qualsiasi tipo, ad esempio, un fiume, un bosco, una catena montuosa.
- Uno spazio, ad esempio un'area di sicurezza che delimita una zona pericolosa, o un'area commerciale fornita di un particolare servizio.
- Un evento che si verifica in una determinata area, ad esempio un incidente stradale verificatosi ad un certo incrocio, oppure una vendita che si svolge in un particolare negozio.

Le caratteristiche sono presenti in ambienti diversi. Ad esempio, gli elementi a cui si è accennato precedentemente, fiume, bosco, catena montuosa, appartengono all'ambiente naturale. Altri elementi, quali città, edifici, uffici, appartengono all'ambiente culturale. I parchi, gli zoo e le fattorie, invece, rappresentano una combinazione tra ambienti naturali e culturali.

In questa pubblicazione, la definizione *informazioni spaziali* si riferisce al tipo di informazioni a cui è possibile accedere utilizzando DB2 Spatial Extender, in altre parole dati e immagini relativi alle caratteristiche geografiche. Alcuni esempi di informazioni geografiche sono:

- La posizione delle caratteristiche geografiche in una cartina (ad esempio, la longitudine e la latitudine che indicano la posizione delle città)
- La posizione di una caratteristica geografica rispetto all'altra (ad esempio, zone della città con ospedali e cliniche oppure edifici abitati prossimi a zone sismiche)
- Il modo in cui le caratteristiche geografiche sono collegate l'una all'altra (ad esempio, i dati che indicano che un determinato sistema di corsi d'acqua si trova in una specifica regione oppure che alcuni ponti di quella regione attraversano gli affluenti di alcuni corsi d'acqua)
- Le misure di una o più caratteristiche geografiche (ad esempio la distanza tra un edificio adibito a uffici e la relativa linea di lotto oppure la lunghezza del perimetro di una riserva di uccelli)

Le informazioni spaziali, anche se combinate con dati relazionali tradizionali, possono risultare utili in attività quali la definizione delle aree per cui si fornisce un servizio, o l'individuazione di luoghi adatti per attività commerciali. Si supponga, ad esempio che un'unità sociale locale debba verificare i richiedenti e i beneficiari di assistenza che vivono attualmente nell'area di applicazione dei

Informazioni su DB2 Spatial Extender

servizi. DB2 Spatial Extender può reperire tali informazioni dall'ubicazione dell'area in questione e dagli indirizzi dei richiedenti e dei beneficiari.

Oppure si supponga che il proprietario di una catena di ristoranti voglia aprire un'attività nelle città vicine. Per stabilire dove è possibile aprire nuovi ristoranti, il proprietario avrà bisogno delle seguenti informazioni: dove si trova una maggiore concentrazione di clientela che di solito frequenta il ristorante, la posizione delle autostrade principali, la zona della città con un tasso di criminalità minore, la posizione dei ristoranti che potrebbero rappresentare eventuali concorrenti. DB2 Spatial Extender e DB2 possono produrre informazioni in grado di soddisfare questo tipo di richiesta. Inoltre, le applicazioni di front-end, anche se non necessarie, possono essere un valido supporto. Utilizzando uno strumento di visualizzazione, ad esempio, è possibile inserire le informazioni prodotte da DB2 Spatial Extender, come la posizione delle concentrazioni di clientela e la vicinanza delle maggiori autostrade a determinati ristoranti, in forma grafica in una carta geografica. Ancora, strumenti di business intelligence sono in grado di inserire in un prospetto informazioni associate, ad esempio nomi e descrizioni di ristoranti in competizione.

Dati spaziali e geodetici

Questa sezione fornisce una panoramica dei dati generati, memorizzati e gestiti per ottenere informazioni spaziali. Gli argomenti trattati sono:

- Modalità di rappresentazione delle caratteristiche geografiche nei dati
- Natura dei dati spaziali e geodetici
- Procedure per la produzione di dati spaziali

Modalità di rappresentazione delle caratteristiche geografiche nei dati

In DB2[®] Spatial Extender, è possibile rappresentare una caratteristica geografica mediante una o più voci di dati, ad esempio le voci di dati della riga di una tabella. Una *voce di dati* è il valore che occupa la cella di una tabella relazionale. Ad esempio, si considerino degli edifici adibiti a ufficio e le residenze. Nella Figura 1, ciascuna riga della tabella BRANCHES rappresenta una filiale della banca. Allo stesso modo, ciascuna riga della tabella CUSTOMERS nella Figura 1 rappresenta un cliente della banca. È possibile considerare il sottoinsieme di ciascuna riga, ovvero le voci di dati che costituiscono l'indirizzo di un cliente, come il domicilio del cliente.

BRANCHES

ID	NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY
937	Airzone-Multern	92467 Airzone Blvd	San Jose	95141	CA	USA

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	95141	CA	USA	A	A

Figura 1. Dati che rappresentano le caratteristiche geografiche. La riga di dati contenuta nella tabella BRANCHES rappresenta una filiale della banca. I dati relativi all'indirizzo nella tabella CUSTOMERS rappresenta l'abitazione di un cliente. I nomi e gli indirizzi riportati nelle tabelle sono fittizi.

Le tabelle della Figura 1 a pagina 4 contengono dati che identificano e descrivono le filiali della banca e i clienti. In questo contesto tali dati vengono definiti *dati aziendali*.

E' possibile convertire una serie secondaria di dati aziendali, ovvero i valori che indicano gli indirizzi delle filiali e dei clienti, in valori da cui generare le informazioni spaziali. Ad esempio, come mostrato nella Figura 1 a pagina 4, l'indirizzo dell'ufficio di una filiale è 92467 Airzone Blvd., San Jose, CA 95141, USA. L'indirizzo di un cliente è 9 Concour Circle, San Jose, CA 95141, USA. DB2 Spatial Extender può convertire questi indirizzi in valori in grado di indicare l'ubicazione della filiale rispetto all'abitazione del cliente. La Figura 2 mostra le tabelle BRANCHES e CUSTOMERS con le nuove colonne che conterranno tali valori.

BRANCHES

ID	NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	95141	CA	USA	

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concour Circle	San Jose	95141	CA	USA		A	A

Figura 2. Tabelle con colonne spaziali aggiunte. In ciascuna tabella, la colonna LOCATION contiene le coordinate corrispondenti agli indirizzi.

Poiché le informazioni spaziali sono ottenute dai dati memorizzati nella colonna LOCATION, nel corso di questa pubblicazione tali dati vengono definiti *dati spaziali*.

Struttura dei dati spaziali

I dati spaziali sono formati da coordinate. Una *coordinata* è un numero che indica:

- Una posizione lungo un'asse relativa ad un'origine, data un'unità di lunghezza, oppure
- Una direzione relativa ad una linea di base o piana, data un'unità di misura angolare.

Ad esempio, la latitudine è una coordinata che denota un angolo relativo al piano equatoriale, generalmente espresso in gradi. La longitudine è una coordinata che denota un angolo relativo al meridiano di Greenwich, anche espresso in gradi. Pertanto, in una carta geografica, la posizione dello Yellowstone National Park è definita dalla latitudine di 44,45 gradi a nord dell'equatore e dalla longitudine di 110,40 gradi ad ovest del meridiano di Greenwich. Più precisamente, tali coordinate si riferiscono al centro dello Yellowstone National Park negli Stati Uniti.

Le definizioni di latitudine e longitudine, i rispettivi punti, le linee e i piani di riferimento, le unità di misura e gli altri parametri associati vengono definiti complessivamente come *un sistema di coordinate*. Esistono anche sistemi di coordinate basati su valori diversi dalla latitudine e dalla longitudine. Tali sistemi di coordinate dispongono di punti, linee e piani di riferimento, unità di misura e altri parametri associati (ad esempio la trasformazione di proiezione) propri. Per ulteriori informazioni, consultare la sezione "Sistemi di coordinate" a pagina 59.

Informazioni su DB2 Spatial Extender

I dati spaziali più semplici consistono in una coppia di singole coordinate che definisce una singola posizione geografica. Dati spaziali più complessi possono essere formati da diverse coordinate che definiscono un percorso lineare delineato, ad esempio, da un fiume o da una strada. Esiste ancora un terzo tipo che comprende le coordinate che definiscono i limiti di un'area; ad esempio, i limiti dei confini di un terreno o di una pianura soggetta a inondazioni periodiche.

Ciascuna rappresentazione di dati spaziali corrisponde a un'istanza di un tipo di dati spaziali. Il tipo di dati per le coordinate che indicano un'ubicazione è ST_Point; il tipo di dati per le coordinate che definiscono percorsi lineari è ST_LineString; il tipo di dati per le coordinate che definiscono i confini di un'area è ST_Polygon. Questi ed altri tipi di dati spaziali, sono tipi strutturati appartenenti a una singola gerarchia.

Natura dei dati geodetici

DB2 Geodetic Extender utilizza gli stessi tipi di dati e funzioni di Spatial Extender per memorizzare i dati geografici in un database DB2. I dati geodetici sono dati spaziali espressi in coordinate di latitudine longitudine, in un sistema di coordinate (consultare la sezione "Sistemi di coordinate" a pagina 59) che descrive una superficie sferica, continua e chiusa. A differenza di Spatial Extender, che considera la Terra come una superficie piana, Geodetic Extender considera la Terra come un globo senza margini o unioni all'altezza dei poli o alla linea del cambio di data. Una carta geografica piana richiede la proiezione delle coordinate per poter trasformare le coordinate sferiche in planari. Geodetic Extender, invece, utilizza la latitudine e la longitudine su un modello ellissoidale della superficie della Terra. Il calcolo delle intersezioni di linea, sovrapposizione di aree, distanza ed area risulteranno accurati e precisi a prescindere dalla posizione.

Per ulteriori informazioni, consultare le sezioni "Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender" a pagina 158 e "Differenze nell'uso di rappresentazioni della Terra piana o sferica" a pagina 197.

Provenienza dei dati spaziali

Come si ottengono i dati spaziali:

- Generandoli dai dati aziendali
- Generandoli dalle caratteristiche spaziali
- Importandoli da fonti esterne

Utilizzo dei dati aziendali come dati di origine

DB2 Spatial Extender è in grado di generare dati spaziali da dati aziendali, ad esempio un indirizzo (come accennato nella sezione "Modalità di rappresentazione delle caratteristiche geografiche nei dati" a pagina 4). Questo processo è denominato *geocoding*. Per osservare la sequenza di trasformazione, vedere la Figura 2 a pagina 5 come immagine "precedente" e la Figura 3 a pagina 7 come immagine "successiva". La Figura 2 a pagina 5 mostra la tabella BRANCHES e la tabella CUSTOMERS che contengono una colonna per i dati spaziali. Tali colonne vengono utilizzate da DB2 Spatial Extender per inserirvi i risultati della funzione di geocoding eseguita su alcuni indirizzi contenuti nelle tabelle per ottenere le rispettive coordinate. La Figura 3 a pagina 7 mostra questo risultato.

BRANCHES

ID	NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	95141	CA	USA	1653 3094

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	95141	CA	USA	953 1527	A	A

Figura 3. Tabelle che includono dati spaziali derivati da dati di origine. La colonna LOCATION nella tabella CUSTOMERS, contiene le coordinate ottenute dall'indirizzo riportato dalle colonne ADDRESS, CITY, POSTAL CODE, STATE_PROV e COUNTRY. Allo stesso modo, la colonna LOCATION nella tabella BRANCHES, contiene le coordinate ottenute dall'indirizzo riportato dalle colonne ADDRESS, CITY, POSTAL CODE, STATE_PROV e COUNTRY di questa tabella.

Per convertire i dati aziendali in coordinate che consentano alle funzioni spaziali di utilizzare tali dati, DB2 Spatial Extender utilizza l'utilità *geocoder*.

Utilizzo delle funzioni per generare i dati spaziali

Oltre al geocoder, esistono altre funzioni in grado di generare dati spaziali. Alcune di queste funzioni, definite *costruttori*, sono in grado di generare dati spaziali dai valori forniti come input, altre invece richiedono che l'input sia costituito da dati spaziali esistenti. Si supponga ad esempio che nella tabella BRANCHES siano definite tutte le filiali di una banca e che la banca voglia sapere il numero di clienti ubicati nel raggio di cinque miglia da ciascuna filiale. Prima che il database sia in grado di fornire questa informazione, è necessario definire l'area che si sviluppa nel raggio di cinque miglia dalle diverse filiali. La funzione ST_Buffer di DB2 Spatial Extender è in grado di creare tale definizione. ST_Buffer, utilizzando le coordinate di ciascuna filiale come input, genera le coordinate che delineano i perimetri delle zone. La Figura 4 mostra la tabella BRANCHES con le informazioni fornite da ST_Buffer.

BRANCHES

ID	NAME	ADDRESS	CITY	POSTAL CODE	STATE_PROV	COUNTRY	LOCATION	SALES_AREA
937	Airzone-Multern	92467 Airzone Blvd	San Jose	95141	CA	USA	1653 3094	1002 2001, 1192 3564, 2502 3415, 1915 3394, 1002 2001

Figura 4. Tabella che include nuovi dati spaziali ottenuti da dati spaziali esistenti. Le coordinate della colonna SALES_AREA sono ottenute mediante la funzione ST_Buffer dalle coordinate della colonna LOCATION. Come per la colonna LOCATION, le colonne presenti in SALES_AREA non sono reali, ma fittizie.

ST_Buffer non è l'unica funzione fornita da DB2 Spatial Extender in grado di ottenere nuovi dati spaziali da dati spaziali esistenti. Per ulteriori informazioni su queste funzioni, consultare la sezione "Concetti correlati" e "Riferimenti correlati" al termine di questo argomento.

Importazione di dati spaziali

E' possibile importare i dati spaziali da file forniti da origini dati esterne. Generalmente questi file contengono dati che vengono applicati alle carte geografiche: reti stradali, segnalazione di zone soggette a inondazioni, zone sismiche, e così via. Utilizzando questi dati in combinazione con i dati spaziali prodotti dall'utente, sarà possibile incrementare la quantità di informazioni spaziali disponibili. Ad esempio per determinare i rischi che potrebbero verificarsi per una

Informazioni su DB2 Spatial Extender

comunità di abitanti, un ente per i lavori pubblici potrà utilizzare ST_Buffer per definire la zona circostante la comunità. L'ente di lavori pubblici potrà quindi importare i dati relativi alle zone soggette a inondazioni e alle zone sismiche per determinare se la zona è a rischio.

Argomenti correlati:

- “DB2 Geodetic Extender” a pagina 157
- “Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender” a pagina 158
- “Funzioni spaziali” a pagina 295
- “Sistemi di coordinate” a pagina 59
- “Funzioni che creano nuove forme geometriche con diverse configurazioni di spazio” a pagina 326

Riferimenti correlati:

- “Differenze nell'uso di rappresentazioni della Terra piana o sferica” a pagina 197
- “Funzioni spaziali supportate da DB2 Geodetic Extender” a pagina 206
- “ST_Buffer” a pagina 354
- “Funzioni che generano nuove forme geometriche da forme geometriche esistenti” a pagina 325

Interazione tra funzioni, informazioni spaziali, dati spaziali e forme geometriche

Questa sezione contiene alcuni concetti su cui si basano le operazioni che è possibile eseguire con DB2[®] Spatial Extender: caratteristiche geografiche, informazioni spaziali, dati spaziali e forme geometriche.

DB2 Spatial Extender consente di ricavare informazioni e figure relative ad oggetti che è possibile definire geograficamente, in base alla rispettiva posizione sulla Terra o in un'area. La documentazione DB2 definisce tali elementi come *informazioni spaziali*, e *caratteristiche geografiche* (in breve, *caratteristiche*).

Ad esempio, è possibile utilizzare DB2 Spatial Extender per determinare se una determinata area abitata, si sovrappone ad una zona proposta per una discarica. Le aree abitate e la zona proposta, sono caratteristiche. Una ricerca di una eventuale sovrapposizione sarebbe un esempio di informazione spaziale. Se viene rilevata una sovrapposizione tra le due aree, l'estensione del rilevamento costituirebbe un'altra informazione spaziale.

Per produrre le informazioni spaziali, DB2 Spatial Extender dovrà elaborare i dati che definiscono la posizione delle caratteristiche. Tali dati, definiti *dati spaziali*, consistono in coordinate che fanno riferimento alle posizioni in una carta geografica o in una proiezione simile. Ad esempio, per determinare se una caratteristica si sovrappone ad un'altra, DB2 Spatial Extender dovrà stabilire la posizione delle coordinate di una caratteristica rispetto alle coordinate dell'altra.

Nell'ambito della tecnologia spaziale, è molto comune rappresentare le caratteristiche geografiche con simboli definiti *forme geometriche*. Le forme geometriche sono in parte visive e in parte matematiche. Consideriamo l'aspetto visivo. Il simbolo di una caratteristica con larghezza e latitudine definite, ad esempio un parco o una città, sarà una figura a più lati. Tale forma geometrica è definita *poligono*. Il simbolo di una caratteristica lineare, ad esempio un fiume o una strada, è una riga. Tale forma geometrica è definita *linea*.

Una forma geometrica dispone di proprietà relative alle informazioni sulla caratteristica che rappresenta. La maggior parte di queste proprietà può essere espressa matematicamente. Ad esempio, le coordinate di una funzione rappresentano una delle proprietà della forma geometrica corrispondente alla caratteristica. Un'altra proprietà, definita *dimensione*, è un valore numerico che indica se una caratteristica dispone di lunghezza o latitudine.

I dati spaziali e alcune informazioni spaziali, possono essere visualizzati come forme geometriche. Nell'esempio descritto in precedenza, relativo alle aree abitate e alla zona di discarica proposta, i dati spaziali delle aree abitate comprendono le coordinate memorizzate nella colonna della tabella di un database DB2. I dati, per convenzione, vengono considerati come forme geometriche. Dal momento che le aree abitate dispongono sia di larghezza che di latitudine, queste forme geometriche saranno poligoni.

Così come i dati spaziali, alcune informazioni spaziali possono essere visualizzate come forme geometriche. Ad esempio, per stabilire se un'area abitata si sovrappone ad una zona proposta come discarica, DB2 Spatial Extender confronterà le coordinate nel poligono che rappresenta la discarica, con le coordinate dei poligoni che rappresentano le aree abitate. Le informazioni ottenute come risultato, ovvero l'intersezione delle due aree, sono considerate poligoni: forme geometriche con coordinate, dimensioni e altre proprietà.

Introduzione

Capitolo 2. Informazioni sulle forme geometriche

Questo capitolo descrive le entità definite forme geometriche, costituite da coordinate che rappresentano le caratteristiche geografiche. Gli argomenti trattati sono:

- Forme geometriche
- Proprietà delle forme geometriche

Forme geometriche

Il Webster's Revised Unabridged Dictionary definisce la *geometria* come "Quel ramo della matematica che studia le relazioni, le proprietà e le misure dei solidi, delle superfici, delle linee e degli angoli; la scienza che si occupa delle proprietà e delle relazioni delle magnitudo; la scienza che si occupa delle relazioni tra gli spazi" La parola *geometria* è stata utilizzata anche per indicare le caratteristiche geometriche utilizzate dai cartografi per disegnare le carte geografiche. Una definizione astratta di questo nuovo significato della parola geometria, potrebbe essere "un punto o un insieme di punti che rappresenta un elemento sulla Terra".

In DB2[®] Spatial Extender, la definizione *operativa* di forma geometrica è "modello di una caratteristica geografica". Il modello può essere espresso utilizzando le coordinate della caratteristica geografica. Il modello, trasmette le informazioni; ad esempio, le coordinate consentono di identificare la posizione della caratteristica geografica rispetto a dei punti fissi di riferimento. Il modello, inoltre, può essere utilizzato per produrre altre informazioni; ad esempio, la funzione ST_Overlaps è in grado di utilizzare come input le coordinate di due aree vicine e indicare come risultato se le due aree si sovrappongono o meno.

Le coordinate di una caratteristica rappresentata da una forma geometrica, sono definite *proprietà* della forma. Alcuni tipi di forma geometrica dispongono anche di altre proprietà; ad esempio l'area, la lunghezza e il contorno.

Le forme geometriche supportate da DB2 Spatial Extender formano una gerarchia, mostrata nella figura seguente. La gerarchia di forme geometriche è definita dal OpenGIS Consortium, Inc. (OGC) nel documento "OpenGIS Simple Features Specification for SQL". E' possibile istanziare sette membri della gerarchia. Ovvero, possono essere definiti con valori di coordinate specifici e visualizzati come in mostrato nella figura.

Informazioni sulle forme geometriche

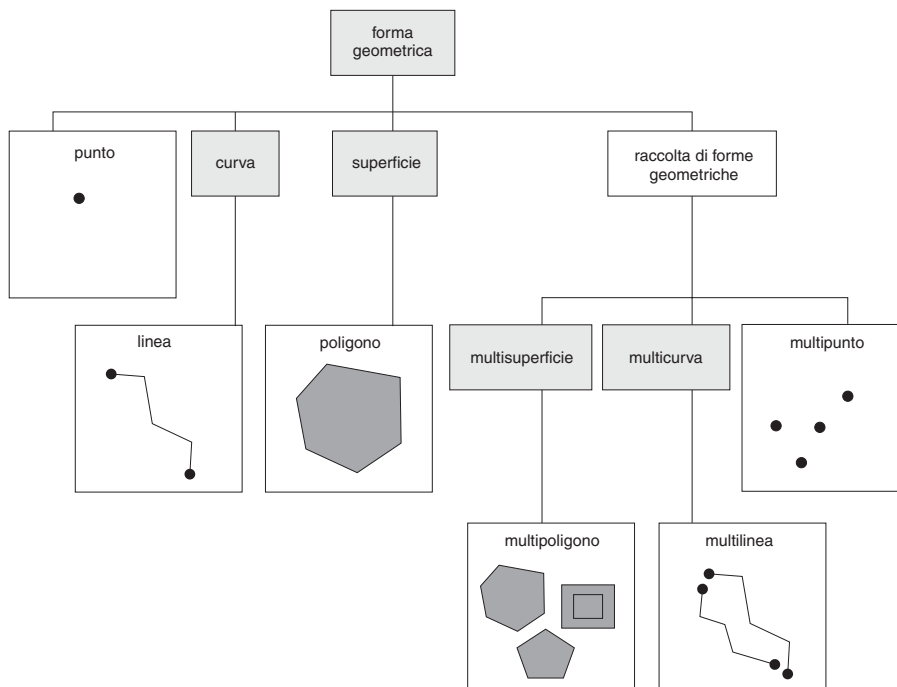


Figura 5. Gerarchia di forme geometriche supportate da DB2 Spatial Extender. La figura contiene le geometrie istanziabili e mostra come possono essere rappresentate visivamente.

I tipi di dati spaziali supportati da DB2 Spatial Extender sono implementazioni delle forme geometriche rappresentate nella figura.

Come indicato, la superclasse *forma geometrica* costituisce l'elemento principale della gerarchia. Il tipo principale ed altri sottotipi validi nella gerarchia non possono essere istanziati. Gli utenti possono eventualmente definire tipi istanziabili e non, personalizzati.

I tipi secondari sono suddivisi in due categorie: i tipi secondari della forma geometrica di base e l'insieme omogeneo di tipi secondari.

Le forme geometriche di base comprendono:

Punti Un singolo punto. I punti rappresentano le caratteristiche geografiche che occupano il luogo in cui una coordinata est-ovest (ad esempio un parallelo), si interseca con una coordinata nord-sud (ad esempio un meridiano). Si supponga che in una carta geografica mondiale, ciascuna città si trova nell'intersezione tra un parallelo e un meridiano, le città potrebbero essere rappresentate da punti.

Linee Una linea tra due punti. Non deve essere una linea retta. Le linee rappresentano caratteristiche geografiche lineari (ad esempio, strade, canali e condutture).

Poligoni

Un poligono o una superficie nel poligono. I poligoni rappresentano caratteristiche geografiche a più lati (ad esempio, un distretto, foreste e aree selvatiche).

L'insieme omogeneo comprende:

Multipunti

Una raccolta di forme geometriche multipunto. I multipunti rappresentano

caratteristiche geografiche i cui componenti si trovano nelle intersezioni tra coordinate est-ovest e nord-sud (ad esempio, un arcipelago le cui isole sono situate nelle intersezioni di meridiani e paralleli).

Multilinee

Una raccolta di forme geometriche curve multipunto, con più linee. Le multilinee rappresentano caratteristiche geografiche composte da più parti (ad esempio, un sistema fluviale o una rete di autostrade).

Multipoligoni

Una raccolta di forme geometriche di superfici multipunto, con più poligoni. I multipoligoni rappresentano caratteristiche geografiche a più parti costituite da componenti a più lati (ad esempio, l'insieme dei terreni coltivabili in una determinata area o un insieme di laghi).

Gli insiemi omogenei sono insiemi di forme geometriche di base. Oltre a condividere le proprietà della forma geometrica di base, gli insiemi omogenei dispongono di proprietà specifiche.

Argomenti correlati:

- "Dati spaziali e geodetici" a pagina 4

Proprietà delle forme geometriche

In questa sezione vengono descritte le seguenti proprietà delle forme geometriche:

- Il tipo a cui appartiene la forma geometrica
- Le coordinate della forma geometrica
- Spazio interno, contorno e spazio esterno della forma geometrica
- Se la forma geometrica è semplice o non semplice
- Se la forma geometrica è vuota o non vuota
- L'involuppo o l'MBR (minimum bounding rectangle) della forma geometrica
- La dimensione
- L'identificativo del sistema di riferimento spaziale a cui è associata una forma geometrica

Tipo

Ciascuna forma geometrica appartiene ad un tipo della gerarchia di forme geometriche supportate da DB2 Spatial Extender. Per una descrizione della gerarchia di forme geometriche, consultare la sezione "Forme geometriche" a pagina 11. E' possibile definire sette tipi appartenenti alla gerarchia, ovvero punti, linee, poligoni, insiemi di forme geometriche, multipunti, multilinee e multipoligoni, utilizzando valori di coordinate specifici.

Coordinate della forma geometrica

Tutte le forme geometriche sono dotate di almeno una coordinata X e una Y, a meno che non si tratti di forme geometriche vuote, nel qual caso non contengono alcuna coordinata. Inoltre, una forma geometrica può disporre di una o più coordinate Z ed M. Le coordinate X, Y, Z ed M sono rappresentate da numeri di doppia precisione. Di seguito vengono trattati i seguenti argomenti:

- Coordinate X e Y
- Coordinate Z
- Coordinate M

Coordinate X e Y

Il *valore di coordinata X* indica una posizione relativa ad un punto in riferimento ad est o ad ovest. Il *valore di coordinata Y* indica una posizione relativa ad un punto in riferimento a nord o a sud.

Coordinate Z

Ad alcune forme geometriche sono associati valori di altitudine o profondità. Ciascun punto che forma una forma geometrica o una caratteristica geografica, può disporre di una coordinata Z che rappresenta l'altezza o la profondità rispetto alla superficie terrestre.

Coordinate M

Una coordinata M (misura) è un valore che raccoglie le informazioni su una caratteristica geografica, memorizzato insieme alle coordinate che definiscono la posizione della caratteristica. Ad esempio, si supponga che si voglia rappresentare un'autostrada nella propria applicazione. Se si desidera che l'applicazione elabori i valori che indicano le distanze lineari o le pietre miliari, è possibile memorizzare tali valori insieme alle coordinate che definiscono le varie posizioni lungo l'autostrada. Le coordinate M sono rappresentate da numeri di doppia precisione.

Spazio interno, contorno e spazio esterno

Tutte le forme geometriche occupano una posizione nello spazio definita dai rispettivi spazi interni, limitrofi ed esterni. Lo spazio esterno ad una forma geometrica è lo spazio non occupato dalla forma. Il contorno di una forma geometrica fa da interfaccia tra lo spazio interno e l'esterno. Lo spazio interno è lo spazio occupato dalla forma geometrica.

Semplice o non semplice

I valori di alcuni tipi secondari di forma geometrica (linee, multipunti e multilinee) possono essere semplici o non semplici. Una forma geometrica è semplice se rispetta tutte le regole topologiche imposte al tipo secondario; in caso contrario è non semplice. Una linea è semplice se non si interseca con il rispettivo spazio interno. Un multipunto è semplice se nessuno degli elementi che lo compongono occupa lo stesso spazio di coordinate. I punti, le superfici, le multisuperfici e le forme geometriche vuote sono sempre forme di tipo semplice.

Chiusa

Una curva è chiusa quando i punti iniziale e finale coincidono. Una multicurva è chiusa se tutti gli elementi che la compongono sono chiusi. Una circonferenza è una semplice curva chiusa.

Vuote o non vuote

Una forma geometrica è vuota se non contiene alcun punto. L'involuppo, il contorno, lo spazio interno ed esterno di una geometria vuota non sono definiti e verranno rappresentati come valori nulli. Una forma geometrica vuota è sempre semplice. I poligono e i multipoligoni vuoti hanno area 0.

MBR (minimum bounding rectangle)

L'MBR di una forma geometrica è la figura geometrica di delimitazione formata dalle coordinate minime e massime (X,Y). Fatta eccezione per i seguenti casi particolari, gli MBR delle forme geometriche formano un rettangolo di delimitazione.

- L'MBR di qualsiasi punto è il punto stesso, poiché le rispettive coordinate X minime e massime corrispondono così come le coordinate Y minime e massime.
- L'MBR di una linea orizzontale o verticale è una linea rappresentata dal limite (i punti finali) della linea di origine.

Dimensione

Una forma geometrica può avere dimensione -1, 0, 1 o 2. Le dimensioni vengono riportate di seguito:

- 1 E' vuota
- 0 Non ha lunghezza e ha area 0 (zero)
- 1 Ha lunghezza maggiore di 0 (zero) e area 0 (zero)
- 2 Ha un'area maggiore di 0 (zero)

I tipi secondari dei punti e dei multipunti hanno dimensione zero. I punti rappresentano le caratteristiche geografiche dimensionali che possono essere modellate con una tupla singola di coordinate, mentre i sottotipi di multipunti rappresentano i dati che devono essere modellati con una serie di punti.

I tipi secondari delle linee e delle multilinee hanno dimensione uno. Rappresentano tratti di strade, affluenti di corsi d'acqua e altre caratteristiche geografiche che in natura hanno aspetto lineare.

I tipi secondari dei poligoni e dei multipoligoni hanno dimensione due. Le caratteristiche geografiche il cui perimetro racchiude un'area definibile, ad esempio, foreste, lotti di terra e laghi, possono essere rappresentate da poligoni o da multipoligoni.

Identificativo del sistema di riferimento spaziale

L'identificativo numerico di un sistema di riferimento spaziale, determina il sistema utilizzato per rappresentare la forma geometrica.

E' possibile accedere a tutti i sistemi di riferimento spaziali noti al database attraverso la vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS.

Introduzione

Capitolo 3. Uso di DB2 Spatial Extender

Uso di DB2 Spatial Extender

Per utilizzare DB2® Spatial Extender e sviluppare progetti che utilizzano dati spaziali, è necessario innanzitutto impostare il programma. Questa sezione contiene un'introduzione alle interfacce che è possibile utilizzare per eseguire attività con dati spaziali.

Interfacce di DB2 Spatial Extender e funzionalità associate

DB2 Spatial Extender dispone di diverse interfacce che consentono di impostare il programma e di creare progetti che utilizzano dati spaziali. Queste interfacce sono:

- Il Centro di controllo DB2, un'interfaccia grafica che comprende finestre, notebook e voci di menu che supportano DB2 Spatial Extender.
- Un processore riga comandi (CLP) fornito da DB2 Spatial Extender. E' definito *CLP db2se*.
- Programmi applicativi che richiamano le procedure memorizzate di DB2 Spatial Extender.

Esistono altre interfacce che consentono di generare informazioni spaziali. Tra queste:

- Query SQL che è possibile avviare dal processore riga comandi DB2, da una finestra query nel centro di controllo DB2, o da un programma applicativo.
- Strumenti di visualizzazione che consentono di trasformare l'informazione spaziale in immagine. Un esempio è ArcExplorer for DB2, creato dall'Environmental Systems Research Institute (ESRI) per IBM®. E' possibile scaricare ArcExplorer for DB2 al sito Web di DB2 Spatial Extender Web:
<http://www.ibm.com/software/data/spatial/>

Impostazione di DB2 Spatial Extender e creazione di progetti

Questa sezione descrive le operazioni da eseguire per impostare DB2 Spatial Extender e per sviluppare progetti che utilizzano dati spaziali e contiene alcuni esempi che illustrano le varie attività. Tali attività si suddividono in due categorie:

- Impostazione di DB2 Spatial Extender
- Creazione di progetti che utilizzano dati spaziali

Impostazione di DB2 Spatial Extender:

Questa sezione descrive come impostare DB2 Spatial Extender servendosi di un esempio in cui un'azienda fittizia esegue le varie operazioni.

Per impostare DB2 Spatial Extender, procedere come segue:

1. Pianificare le operazioni di base (stabilire i progetti da sviluppare, l'interfaccia o le interfacce da utilizzare, selezionare il personale che dovrà gestire DB2 Spatial Extender e sviluppare i progetti, e così via).

Esempio: L'ambiente di sistema della compagnia di assicurazioni Safe Harbor Real Estate comprende un sistema DB2 Universal Database™ e un file di sistema distinto per la gestione dei dati spaziali. I risultati delle interrogazioni possono includere combinazioni di dati provenienti dai due sistemi. Ad

esempio, in una tabella DB2 sono memorizzate le informazioni relative ai ricavi e un file del file system contiene l'ubicazione delle filiali della compagnia. In questo modo, è possibile individuare le filiali che portano alla compagnia specifiche quantità di ricavi e determinare la posizione di tali filiali. I dati dei due sistemi non sono integrabili (ad esempio, gli utenti non possono unire le colonne DB2 ai record del file system, e i servizi DB2, in questo caso l'ottimizzazione delle interrogazioni, non sono disponibili per il file system). Per risolvere questo problema, la compagnia Safe Harbor acquista DB2 Spatial Extender Versione 8 e crea un nuovo reparto di sviluppo spaziale (in breve, reparto spaziale).

Il primo compito del nuovo reparto è di integrare DB2 Spatial Extender nell'ambiente DB2 di Safe Harbor:

- Il team di gestione del reparto assegna al team di gestione spaziale le operazioni di installazione e implementazione di DB2 Spatial Extender e al team di analisi spaziale la generazione e l'analisi delle informazioni spaziali.
- Il team di gestione dispone di un potente sistema UNIX[®], quindi per gestire DB2 Spatial Extender, viene deciso di utilizzare il processore riga comandi db2se.
- Poiché le decisioni aziendali di Safe Harbor vengono adattate alle esigenze dei clienti, il team di gestione decide di installare DB2 Spatial Extender nel database contenente le informazioni relative ai clienti. La maggior parte delle informazioni vengono memorizzate nella tabella CUSTOMERS.

2. Installare DB2 Spatial Extender.

Esempio: Il team di gestione spaziale installa DB2 Spatial Extender su una macchina UNIX in ambiente DB2.

3. Se si dispone già della versione 7 di DB2 Spatial Extender, migrare i dati alla versione 8.

Esempio: La versione 8 è il primo rilascio acquistato dalla compagnia Safe Harbor. Non è necessario eseguire la migrazione.

4. Configurare il database in modo che possa ricevere i dati spaziali. E' necessario modificare i parametri di configurazione in modo da garantire memoria e spazio sufficiente per le funzioni spaziali, i file di registrazione e le applicazioni di DB2 Spatial Extender.

Esempio: Un membro del team di gestione spaziale imposta le caratteristiche del file di registrazione della transazione, la dimensione heap dell'applicazione e del controllo dell'applicazione, in base ai valori richiesti da DB2 Spatial Extender.

5. Impostare le risorse spaziali per il database. Tali risorse includono un catalogo di sistema, tipi di dati spaziali, funzioni spaziali, un geocoder, ed altri oggetti. L'attività di impostazione di tali risorse viene indicata come *abilitazione del database per le operazioni spaziali*.

Il geocoder fornito converte gli indirizzi degli Stati Uniti in dati spaziali. E' chiamato DB2SE_USA_GEOCODER. L'organizzazione dell'utente o altre organizzazioni possono fornire geocoder che convertono indirizzi non appartenenti agli Stati Uniti e altri tipi di dati in dati spaziali.

Esempio: Il team di gestione spaziale imposta le risorse necessarie ai progetti in corso.

- Un membro del team richiama le risorse che abilitano il database per le operazioni spaziali lanciando un comando. Queste risorse comprendono il catalogo di DB2 Spatial Extender, tipi di dati spaziali, funzioni spaziali e così via.

- Poiché la compagnia Safe Harbor vuole estendere la propria attività in Canada, il team di gestione spaziale richiede ai produttori canadesi i geocoder per la conversione di indirizzi canadesi in dati spaziali. La Safe Harbor non prevede di acquistare questi geocoder prima di alcuni mesi, quindi le prime ubicazioni in base alle quali verranno raccolti i dati saranno negli Stati Uniti.

Creazione di progetti che utilizzano dati spaziali:

Dopo aver impostato DB2 Spatial Extender, è possibile avviare l'elaborazione di progetti che si basano su dati spaziali. Questa sezione descrive le attività coinvolte nella creazione di tali progetti rifacendosi all'esempio della compagnia di assicurazioni Safe Harbor Real Estate Insurance Company iniziato nella sezione precedente.

Per creare un progetto basato su dati spaziali, procedere come segue:

1. Pianificare le operazioni di base (impostare gli obiettivi del progetto, stabilire quali sono le tabelle e i dati necessari, stabilire il sistema di coordinate o i sistemi da utilizzare, e così via).

Esempio: Il reparto si prepara allo sviluppo del progetto:

- Il team di gestione imposta gli obiettivi del progetto:
 - Stabilire dove collocare le nuove filiali
 - Assegnare i premi in base alla vicinanza dei clienti alle aree di rischio (aree con un numero elevato di incidenti stradali, aree ad alto tasso di criminalità, zone soggette ad alluvioni, zone sismiche e così via)
- Questo progetto riguarda clienti e uffici situati negli Stati Uniti, pertanto il team di gestione spaziale decide di:
 - Utilizzare il sistema di coordinate per gli Stati Uniti fornito da DB2 Spatial Extender, GCS_NORTH_AMERICAN_1983.
 - Utilizzare DB2SE_USA_GEOCODER, perché è stato ideato per eseguire operazioni di geocoding su indirizzi degli Stati Uniti.
- Il team di gestione spaziale stabilisce i dati necessari per il raggiungimento degli obiettivi del progetto e le tabelle che conterranno i risultati.

2. Se necessario, creare un sistema di coordinate.

Esempio: Poiché la Safe Harbor ha deciso di utilizzare GCS_NORTH_AMERICAN_1983, la compagnia può ignorare questo punto.

3. Decidere se il sistema di riferimento spaziale esistente è adatto alle proprie necessità. In caso contrario crearne uno.

Un *sistema di riferimento spaziale* è una serie di valori che comprende:

- Le coordinate che definiscono l'estensione massima possibile di uno spazio, in base a un intervallo di coordinate dato. È necessario stabilire il valore massimo dell'intervallo di coordinate che è possibile rilevare dal sistema di coordinate utilizzato, e selezionare o creare un sistema di riferimento spaziale che rifletta tale intervallo.
- Il nome del sistema di coordinate dal quale provengono le coordinate.
- I numeri utilizzati nelle operazioni matematiche per convertire le coordinate ricevute come input, in valori che è possibile elaborare. Le coordinate vengono memorizzate nel formato convertito e restituite all'utente nel formato originale.

Esempio: DB2 Spatial Extender dispone del sistema di riferimento spaziale NAD83_SRS_1, da utilizzare con GCS_NORTH_AMERICAN_1983. Il team di gestione spaziale decide di utilizzare NAD83_SRS_1.

4. Creare le colonne spaziali, in base alle proprie impostazioni. Se i dati devono essere letti da uno strumento di visualizzazione, è necessario che nella tabella o nella vista vi sia una sola colonna spaziale. Oppure, se la colonna è inclusa in una tabella che contiene più colonne spaziali, inserirla in una vista che non contiene altre colonne spaziali, in modo che lo strumento di visualizzazione sia in grado di leggere i dati in essa contenuti.

Esempio: Il team di gestione spaziale definisce le colonne che contengono dati spaziali.

- Il team aggiunge la colonna LOCATION alla tabella CUSTOMERS. La tabella contiene già gli indirizzi dei clienti. DB2SE_USA_GEOCODER converte gli indirizzi in dati spaziali e DB2 li memorizza nella colonna LOCATION.
- Il team crea le tabelle OFFICE_LOCATIONS e OFFICE_SALES in cui vengono inseriti i dati che adesso risultano memorizzati in un file system separato. Questi dati includono gli indirizzi delle filiali della compagnia Safe Harbor, i dati spaziali ottenuti da questi indirizzi mediante il geocoder e i dati spaziali che definiscono l'area che circonda ciascuna filiale nel raggio di cinque miglia. I dati ottenuti dal geocoder vengono inseriti nella colonna LOCATION della tabella OFFICE_LOCATIONS e i dati che definiscono le aree nella colonna SALES_AREA nella tabella OFFICE_SALES.

5. Impostare le colonne spaziali in modo che possano essere lette da strumenti di visualizzazione. Per eseguire questa operazione è necessario registrare le colonne nel catalogo di DB2 Spatial Extender. Dopo aver registrato una colonna spaziale, DB2 Spatial Extender vincola i dati in modo che vengano considerati come appartenenti allo stesso sistema di riferimento spaziale. Tale vincolo assicura l'integrità dei dati ed è obbligatorio per molti strumenti di visualizzazione.

Esempio: Il team di gestione spaziale prevede di utilizzare degli strumenti di visualizzazione per inserire in una carta geografica in formato grafico i dati contenuti nelle colonne LOCATION e SALES_AREA. Il team quindi registra le tre colonne.

6. Inserire i dati nelle colonne spaziali:

Importare i dati relativi ai progetti che richiedono l'importazione di dati spaziali.

Per i progetti che richiedono l'utilizzo di un geocoder, procedere come segue:

- Impostare le informazioni di controllo nel momento in cui il geocoder viene richiamato.
- E' possibile impostare il geocoder in modo che venga eseguito automaticamente ogni qual volta un nuovo indirizzo viene aggiornato o aggiunto al database.

Eseguire il geocoder in modalità batch, se necessario.

Se il progetto richiede che i dati spaziali vengano creati utilizzando una funzione spaziale, eseguire la funzione.

Esempio: Il team di gestione spaziale inserisce i dati nella colonna LOCATION della tabella CUSTOMER, nelle tabelle OFFICE_LOCATION, OFFICE_SALES e nella nuova tabella HAZARD_ZONES:

- Il team esegue il geocoding degli indirizzi nella tabella CUSTOMER mediante DB2SE_USA_GEOCODER. Le coordinate prodotte dall'operazione di geocoding vengono inserite nella colonna LOCATION della tabella.
- Il team utilizza un'utilità per caricare i dati relativi alle filiali dal file system in un file. A questo punto, importa i dati nella nuova tabella OFFICE_LOCATIONS.

- Il team crea una tabella HAZARD_ZONES, registra le colonne spaziali appartenenti alla tabella e vi importa i dati. I dati provengono da un file forniti da un produttore di carte geografiche.
7. Facilitare l'accesso alle colonne spaziali, se necessario. In particolare definire gli indici che consentono a DB2 di accedere rapidamente ai dati spaziali e definire le viste che consentono agli utenti di richiamare i dati. Se si desidera accedere alle colonne spaziali delle viste utilizzando degli strumenti di visualizzazione, è necessario registrare tali colonne con DB2 Spatial Extender.

Esempio: Il team di gestione spaziale crea degli indici per le colonne registrate e crea una vista che unisce le colonne delle tabelle CUSTOMERS e HAZARD ZONES. Successivamente registra le colonne spaziali nella vista.

8. Generare informazioni spaziali e informazioni aziendali correlate. Analizzare le informazioni. Questa attività prevede l'interrogazione delle colonne spaziali e delle colonne non spaziali correlate. In questo tipo di interrogazione, è possibile includere funzioni di DB2 Spatial Extender che restituiscono un'ampia varietà di informazioni; ad esempio le coordinate di una zona di sicurezza che circonda una zona a rischio, oppure la distanza minima tra questa zona e l'edificio pubblico più vicino.

Esempio: Il team di analisi spaziale esegue delle interrogazioni per ottenere informazioni che consentono di raggiungere gli obiettivi originari: stabilire la collocazione delle nuove filiali e regolare i premi sulla base della vicinanza dei clienti alle zone a rischio.

Attività correlate:

- "Impostazione e installazione di Spatial Extender" a pagina 25
- "Abilitazione di un database per le operazioni spaziali" a pagina 54
- "Registrazione di un geocoder" a pagina 56
- "Configurazione di un database per l'inserimento dei dati spaziali" a pagina 49
- "Importazione dei dati shape in una tabella nuova o esistente" a pagina 88
- "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90
- "Impostazione delle funzioni di geocoding" a pagina 94
- "Impostazione dell'esecuzione automatica del geocoder" a pagina 96
- "Esecuzione del geocoder in modalità batch" a pagina 97
- "Esportazione dei dati in un file di trasferimento SDE" a pagina 92
- "Selezione o creazione dei sistemi di coordinate" a pagina 66
- "Registrazione delle colonne spaziali" a pagina 85
- "Creazione di colonne spaziali" a pagina 83
- "Creazione degli indici di griglia spaziali" a pagina 106
- "Richiamo delle procedure memorizzate di DB2 Spatial Extender da un'applicazione" a pagina 134
- "Inserimento del file di intestazione di DB2 Spatial Extender nelle applicazioni spaziali" a pagina 133

Riferimenti correlati:

- "Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti" a pagina 125

Parte 2. Impostazione di DB2 Spatial Extender

Capitolo 4. Introduzione a DB2 Spatial Extender

Questo capitolo contiene le istruzioni per l'installazione e la configurazione di Spatial Extender per AIX, HP-UX, Windows NT, Windows 2000, Linux, Linux per z/OS e Solaris Operating Environment. Inoltre contiene informazioni sulla risoluzione di eventuali problemi di installazione e configurazione che potrebbero verificarsi all'avvio di Spatial Extender.

Impostazione e installazione di Spatial Extender

Questa sezione descrive i seguenti argomenti:

- Requisiti per l'installazione di Spatial Extender
- Istruzioni di installazione per le piattaforme UNIX e Windows
- Istruzioni per la creazione di un'istanza DB2 Spatial Extender
- Verifica dell'installazione di Spatial Extender
- Programma di esempio Suggerimenti per la risoluzione dei problemi durante l'installazione

Impostazione e installazione di Spatial Extender

Un sistema DB2 Spatial Extender comprende DB2 Universal Database, DB2 Spatial Extender e, per la maggior parte delle applicazioni, un geobrowser. L'uso del geobrowser non è obbligatorio, ma può rivelarsi molto utile per visualizzare i risultati delle interrogazioni spaziali in formato di carta geografica. I database abilitati per le operazioni spaziali si trovano sul server. E' possibile utilizzare applicazioni client per accedere a dati spaziali attraverso procedure memorizzate e interrogazioni spaziali di DB2 Spatial Extender. E' anche possibile configurare DB2 Spatial Extender in un ambiente autonomo, che rappresenta una configurazione in cui sia il client che il server sono ubicati sulla stessa macchina. In entrambe le configurazioni, client-server e autonoma, è possibile visualizzare dati spaziali mediante un geobrowser, ad esempio ArcExplorer for DB2 o la suite di strumenti ArcGIS di ESRI eseguita con ArcSDE.

E' possibile scaricare ArcExplorer for DB2 gratuitamente al sito Web di IBM DB2 Spatial Extender:

<http://www.ibm.com/software/data/spatial/>

Prerequisiti:

Prima di impostare DB2 Spatial Extender, è necessario che il software DB2 sia installato e configurato sul client e sul server.

Procedura:

1. Verificare che il sistema disponga di tutti i requisiti software.
2. Installare Spatial Extender. Le operazioni da eseguire variano in base al sistema operativo utilizzato:
 - Windows
 - AIX
 - HP-UX

- Solaris Operating Environment
 - Linux
3. Piattaforme UNIX: Creare un'istanza di DB2 Spatial Extender.
 4. Verificare l'esito dell'installazione. Verificare l'esito dell'installazione.
 5. Se si verificassero problemi, consultare la sezione risoluzione dei problemi e procedere come indicato.
 6. Se si desidera accedere alla documentazione DB2 dal proprio computer e il Centro informazioni DB2 non è ancora stato installato, consultare la sezione Installazione del Centro di informazioni DB2 (UNIX) o Installazione del Centro di informazioni DB2 (Windows). Il Centro di informazioni DB2 contiene la documentazione relativa a DB2 Universal Database e i prodotti DB2 correlati.

Argomenti correlati:

- "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26

Attività correlate:

- "Installazione di DB2 Spatial Extender per Windows" a pagina 27
- "Installazione di DB2 Spatial Extender per AIX" a pagina 29
- "Installazione di DB2 Spatial Extender per HP-UX" a pagina 31
- "Installazione di DB2 Spatial Extender per Solaris Operating Environment" a pagina 33
- "Installazione di DB2 Spatial Extender per Linux" a pagina 35
- "Creazione dell'istanza di DB2 Spatial Extender" a pagina 37
- "Verifica dell'installazione di Spatial Extender" a pagina 39
- "Risoluzione dei problemi di installazione" a pagina 40
- "Installazione del Centro informazioni DB2 utilizzando il Wizard di installazione DB2 (Unix)" nella pubblicazione *Infrastructure Topics (DB2 Common Files)*
- "Installazione del Centro informazioni DB2 utilizzando il Wizard di installazione DB2 (Windows)" nella pubblicazione *Infrastructure Topics (DB2 Common Files)*
- "Download di ArcExplorer per DB2" a pagina 41

Riferimenti correlati:

- "CD per dati e carte geografiche di DB2 Spatial Extender" a pagina 42

Requisiti di sistema per l'installazione di Spatial Extender

Prima di installare DB2[®] Spatial Extender, assicurarsi che il sistema disponga del software e dello spazio necessari per l'installazione.

Sistemi operativi:

E' possibile installare DB2 Spatial Extender sulle versioni a 32-bit e a 64-bit di Windows[®], AIX[®], HP-UX, Solaris e Linux per Intel. Spatial Extender supporta Linux per S/390[®] (32-bit), ma non supporta Linux per zSeries[®] (64-bit).

Requisiti software:

Per installare Spatial Extender, è necessario che il seguente software DB2 sia installato e configurato sul server:

Software del server

DB2 Universal Database[™] Enterprise Server Edition Versione 8.2 deve essere

già installato sul sistema *prima* di installare DB2 Spatial Extender. Se si desidera utilizzare il Centro di controllo DB2, creare e configurare DB2 Administration Server (DAS). Per ulteriori informazioni sulla creazione e sulla configurazione di DAS, fare riferimento a *IBM® DB2 Universal Database Administration Guide: Implementation*.

Software per il client spaziale

Se DB2 Spatial Extender viene installato in Windows, l'installazione predefinita comprende il client spaziale. Per DB2 Spatial Extender su AIX, HP-UX, Solaris Operating Environment, Linux per Intel o Linux per S/390, è possibile installare il client spaziale quando si installa il server DB2 con il client di gestione o di sviluppo applicazioni. Se non si desidera installare questi client, sarà necessario installare il client spaziale manualmente scegliendo l'opzione Installazione personalizzata.

Requisiti di spazio su disco:

Per installare Spatial Extender, occorre soddisfare i requisiti di spazio su disco elencati nella seguente tabella. Il codice di libreria per DB2 Spatial Extender integra il codice per DB2 Geodetic Extender ma non la chiave di licenza.

Tabella 1. Requisiti di spazio su disco per DB2 Spatial Extender

Software di DB2 Spatial Extender	Spazio su disco
Software del server per Spatial Extender:	596 MB di spazio su disco:
<ul style="list-style-type: none"> Codice della libreria server di Spatial Extender e Geodetic Extender, dati di riferimento del geocoder di esempio e documentazione 	<ul style="list-style-type: none"> 33 MB
<ul style="list-style-type: none"> Facoltativo e disponibile su un CD a parte: dati di riferimento del geocoder (Stati Uniti) 	<ul style="list-style-type: none"> 563 MB

La Tabella 1 indica lo spazio su disco richiesto per l'installazione tipica di DB2 Universal Database e DB2 Spatial Extender in ambiente Windows oppure con i componenti preselezionati in su AIX, HP-UX, Solaris Operating Environment, Linux per Intel e Linux per S/390. Se si desidera eseguire un diverso tipo di installazione di DB2 Spatial Extender o se è già stato eseguito un altro tipo di installazione di DB2 Universal Database, i requisiti di spazio su disco subiranno delle variazioni.

Una volta soddisfatti tutti i requisiti software e di spazio su disco, sarà possibile procedere con l'installazione di Spatial Extender.

Installazione di DB2 Spatial Extender per Windows

L'impostazione di DB2 Spatial Extender prevede una prima fase di installazione.

Per installare DB2 Spatial Extender su sistemi operativi Windows è possibile utilizzare il wizard per l'installazione DB2 o un file di risposte.

Suggerimento: per installare Spatial Extender è preferibile utilizzare il wizard di installazione DB2 che dispone di un'interfaccia grafica di semplice utilizzo e di una guida per l'installazione, crea automaticamente utente e gruppi, e provvede alla configurazione dei protocolli e alla creazione dell'istanza.

Se si sceglie di utilizzare il wizard, è possibile interrompere il processo di installazione in qualsiasi momento scegliendo il pulsante **Annulla**.

Prerequisiti:

Per installare DB2 Spatial Extender, è necessario che il prodotto DB2 Versione 8 server sia già installato.

Procedura:

Per installare Spatial Extender per Windows utilizzando il wizard di installazione DB2, procedere come segue:

1. Inserire il CD di Spatial Extender nell'unità CD. Verrà aperta l'utilità Launchpad di installazione DB2, un'interfaccia da cui è possibile installare DB2 Spatial Extender.
2. Scegliere **Installa prodotti**.
3. Selezionare DB2 Spatial Extender e scegliere **AVANTI**. Viene aperto il wizard di installazione DB2. Scegliere **AVANTI**. Attenersi alle istruzioni indicate nel wizard per completare l'installazione. Se durante l'installazione si desiderano informazioni, è possibile avviare in qualsiasi momento la guida in linea scegliendo ?.

Per installare Spatial Extender per Windows utilizzando un file di risposte, procedere come segue:

1. Collegarsi al sistema utilizzando l'account utente con il quale si desidera eseguire l'installazione.
2. Inserire il CD di Spatial Extender. Per ulteriori informazioni, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.
3. Lanciare i seguenti comandi per eseguire il programma di installazione:

Comando db2setup

```
db2setup -f -i lingua -l file_di_registrazione  
-t file_di_traccia -u file_di_risposte -h
```

Dove:

- f Forza l'interruzione dei processi DB2 prima dell'installazione.
- i (*lingua*)
Corrisponde alle due lettere del codice lingua in cui eseguire l'installazione.
- l (*file_di_registrazione*)
Il percorso completo e il nome del file di registrazione da utilizzare.
- t (*file_di_traccia*)
Genera un file di traccia con le informazioni relative all'installazione.
- u (*file_di_risposte*)
Indica il nome completo del file di risposte. Se il file di risposte di esempio è stato modificato e ridenominato, assicurarsi che questo parametro corrisponda al nuovo nome. Il parametro è obbligatorio. Il

file di risposte si trova nella directory
db2\Windows\samples\db2gse.rsp nel CD di installazione di DB2
Spatial Extender.

-?, -h Genera informazioni sull'utilizzo.

- Al termine dell'installazione, controllare i messaggi nel file di registrazione.

Argomenti correlati:

- “Requisiti di sistema per l'installazione di Spatial Extender” a pagina 26

Attività correlate:

- “Creating and editing a response file (Windows)” nella *Installation and Configuration Supplement*
- “Verifica dell'installazione di Spatial Extender” a pagina 39
- “Risoluzione dei problemi di installazione” a pagina 40

Installazione di DB2 Spatial Extender per AIX

Per installare DB2 Spatial Extender per AIX è possibile utilizzare il wizard per l'installazione DB2, lo script db2_install oppure l'interfaccia SMIT (System Management Interface Tool).

Suggerimento: per installare Spatial Extender è preferibile utilizzare il wizard di installazione DB2 che dispone di un'interfaccia grafica di semplice utilizzo e di una guida per l'installazione, crea automaticamente utente e gruppi, e provvede alla configurazione dei protocolli e alla creazione dell'istanza. Se si decide di non utilizzare il wizard, è possibile installare Spatial Extender mediante lo script db2_install o utilizzando l'interfaccia SMIT (System Management Interface Tool) di AIX. Si consiglia di utilizzare SMIT agli utenti esperti, solo se è necessario eseguire diverse operazioni di impostazione manuali durante il processo di installazione.

Prerequisiti:

Prima di avviare il processo di installazione di Spatial Extender su AIX:

- Assicurarsi che il sistema disponga del software richiesto e di memoria e spazio su disco sufficienti.
- Aggiornare i parametri di configurazione e riavviare il sistema per tutti i client e i server DB2 con sistema operativo AIX.
- Se si desidera eseguire l'installazione su un server o in un ambiente autonomo, è necessario che il prodotto DB2 Versione 8 server sia già installato.

Nota: Verificare che DB2 Spatial Client sia già installato. Il client DB2 Spatial Extender e i componenti di esempio sono integrati nelle installazioni del client e del server DB2. E' possibile installare i componenti spaziali mediante l'installazione personalizzata di DB2, selezionando la funzione **Spatial Extender Client in Supporto client**, e la funzione **Esempi Spatial Extender in Strumenti di sviluppo applicazioni**. Se si desiderano solo le funzionalità del client spaziale e i componenti spaziali sono già stati installati con DB2, non sarà necessario eseguire la seguente procedura di installazione di DB2 Spatial Extender.

- E' necessario disporre di autorizzazione root.

Procedura:

Introduzione

Per installare Spatial Extender utilizzando il wizard di installazione DB2, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD di Spatial Extender. Verrà aperta l'utilità Launchpad di installazione DB2, un'interfaccia da cui è possibile installare DB2 Spatial Extender. Per informazioni sul mount del CD, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.
3. Selezionare DB2 Spatial Extender e scegliere **AVANTI**.
4. Viene aperto il wizard di installazione DB2. Attenersi alle istruzioni indicate nel wizard per completare l'installazione. Se durante l'installazione si desiderano informazioni, è possibile avviare in qualsiasi momento la guida in linea scegliendo ?.

Per installare DB2 Spatial Extender utilizzando lo script db2_install, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD appropriato.
3. Per avviare lo script, immettere il comando **./db2_install**. Lo script db2_install si trova nella directory root nel CD dei prodotti di DB2 Versione 8. Lo script db2_install richiederà di inserire la parola chiave del prodotto.
4. Per installare DB2 Spatial Extender, immettere **DB2.GSE**.

Per installare DB2 Spatial Extender utilizzando SMIT (System Management Interface Tool), procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD di Spatial Extender.
3. Immettere il comando **smit install_latest**.
4. Digitare /cdrom/db2 nella **INPUT device/directory** per il campo software.
5. Fare clic su **DO** oppure premere Invio per verificare che la directory di installazione esista.
6. Nel campo **Software da installare**, indicare se devono essere installati i componenti client o server.

Nota: Il file ComponentList.htm contenuto nel CD di DB2 Spatial Extender, contiene l'elenco completo dei componenti che è necessario installare per DB2 Spatial Extender.

7. Scegliere **DO** o premere Invio. Viene richiesto di confermare i parametri di installazione indicati.
8. Premere Invio per confermare.
9. Scollegarsi.

Al termine dell'installazione Spatial Extender risulta installato nella directory /usr/opt/db2_08_01 insieme agli altri prodotti DB2.

A questo punto creare l'istanza DB2 e verificare l'esito dell'installazione.

Argomenti correlati:

- "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26

Attività correlate:

- “Installing a DB2 product using SMIT (AIX)” nella *Installation and Configuration Supplement*
- “Mounting the CD-ROM (AIX)” nella *Quick Beginnings for DB2 Servers*
- “Installing a DB2 product using the db2_install script (UNIX)” nella *Installation and Configuration Supplement*
- “Verifica dell’installazione di Spatial Extender” a pagina 39
- “Risoluzione dei problemi di installazione” a pagina 40

Installazione di DB2 Spatial Extender per HP-UX

E’ possibile installare Spatial Extender utilizzando il wizard di installazione DB2[®], mediante lo script db2_install o utilizzando il comando **swinstall**.

Suggerimento: per installare Spatial Extender è preferibile utilizzare il wizard di installazione DB2 che dispone di un’interfaccia grafica di semplice utilizzo e di una guida per l’installazione, crea automaticamente utente e gruppi, e provvede alla configurazione dei protocolli e alla creazione dell’istanza. Se si decide di non utilizzare il wizard, è possibile installare Spatial Extender per HP-UX utilizzando lo script db2_install o il comando **swinstall**. Si consiglia di utilizzare il comando HP-UX **swinstall** agli utenti esperti, solo se è necessario eseguire diverse operazioni di impostazione manuali durante il processo di installazione.

Prerequisiti:

Per installare DB2 Spatial Extender per HP-UX, procedere come segue:

- Assicurarsi che il sistema disponga dell’hardware e del software richiesti e di memoria sufficiente.
- E’ necessario che il prodotto DB2 Versione 8 server sia già installato.

Nota: Verificare che DB2 Spatial Client sia già installato. Il client DB2 Spatial Extender e i componenti di esempio sono integrati nelle installazioni del client e del server DB2. E’ possibile installare i componenti spaziali mediante l’installazione personalizzata di DB2, selezionando la funzione **Spatial Extender Client in Supporto client**, e la funzione **Esempi Spatial Extender in Strumenti di sviluppo applicazioni**. Se si desiderano solo le funzionalità del client spaziale e i componenti spaziali sono già stati installati con DB2, non sarà necessario eseguire la seguente procedura di installazione di DB2 Spatial Extender.

- Aggiornare i parametri di configurazione e riavviare il sistema per tutti i client e i server DB2 con sistema operativo HP-UX.
- E’ necessario disporre di autorizzazione root.

Procedura:

Per installare Spatial Extender per HP-UX utilizzando il wizard di installazione DB2, procedere come segue:

1. Inserire ed eseguire il mount del CD di DB2 Spatial Extender. Verrà aperta l’utilità Launchpad di installazione DB2, un’interfaccia da cui è possibile installare DB2 Spatial Extender.
2. Selezionare DB2 Spatial Extender e scegliere **AVANTI**. Viene aperto il wizard di installazione DB2. Scegliere **AVANTI**. Attenersi alle istruzioni indicate nel

wizard per completare l'installazione. Se durante l'installazione si desiderano informazioni, è possibile avviare in qualsiasi momento la guida in linea scegliendo ?.

Per installare Spatial Extender per HP-UX utilizzando lo script `db2_install`, procedere come segue:

1. Collegarsi come utente con autorizzazione `root`.
2. Inserire ed eseguire il mount del CD appropriato.
3. Per avviare lo script, immettere il comando `./db2_install`. Lo script `db2_install` si trova nella directory `root` nel CD dei prodotti di DB2 Versione 8. Lo script `db2_install` richiederà di inserire la parola chiave del prodotto.
4. Per installare DB2 Spatial Extender, immettere `DB2.GSE`.

Per installare Spatial Extender per HP-UX mediante il comando `swinstall`, procedere come segue:

1. Collegarsi come utente con autorizzazione `root`.
2. Inserire ed eseguire il mount del CD di Spatial Extender.
3. Eseguire il programma `swinstall` immettendo il seguente comando:
`swinstall -x autoselect_dependencies=true`

Viene aperta una finestra per la selezione del software e una finestra per l'indicazione dell'origine. Se necessario, nella finestra per l'indicazione dell'origine, modificare il valore nel campo **Source host name**.

4. Nel campo **Source Depot Path** immettere `/cdrom/db2/hpux` dove `/cdrom` rappresenta la directory di attivazione del CD.
5. Per tornare alla finestra di selezione del software, scegliere **OK**. Questa finestra contiene un elenco dei software disponibili che è possibile installare.
6. Selezionare i prodotti che si desidera installare e per i quali si dispone di apposita licenza.
7. Per selezionare il prodotto da installare, selezionare **Mark for Install** nel menu **Actions**. Viene visualizzato un messaggio:
`In addition to the software you just marked, other software was automatically marked to resolve dependencies. This message will not appear again.`
8. Scegliere **OK**.
9. Per avviare l'installazione del prodotto e per aprire la finestra **Install Analysis**, scegliere **Install (analysis)** dal menu **Actions**.
10. Quando il campo **Status** mostra il messaggio **Ready**, nella finestra **Install Analysis** scegliere **OK**.
11. Per confermare che si desidera installare il software, nella finestra di conferma scegliere **Sì**.

Durante l'installazione del software la finestra di installazione mostrerà lo stato di elaborazione dei dati; quando il campo **Status** indicherà **Ready** viene visualizzata la finestra **Note**. Il programma `swinstall` carica la serie di file ed esegue gli script di controllo.

12. Per uscire dal programma `swinstall`, scegliere **Exit** dal menu **File**.

Al termine dell'installazione, Spatial Extender risulta installato nella directory `/opt/IBM/db2/V8.1` insieme agli altri prodotti DB2.

A questo punto creare l'istanza DB2 e verificare l'esito dell'installazione.

Argomenti correlati:

- “Requisiti di sistema per l’installazione di Spatial Extender” a pagina 26

Attività correlate:

- “Installing a DB2 product using swinstall (HP-UX)” nella *Installation and Configuration Supplement*
- “Mounting the CD-ROM (HP-UX)” nella *Quick Beginnings for DB2 Servers*
- “Installing a DB2 product using the db2_install script (UNIX)” nella *Installation and Configuration Supplement*
- “Verifica dell’installazione di Spatial Extender” a pagina 39
- “Risoluzione dei problemi di installazione” a pagina 40

Installazione di DB2 Spatial Extender per Solaris Operating Environment

E’ possibile installare Spatial Extender utilizzando il wizard di installazione DB2[®], mediante lo script `db2_install` o utilizzando il comando `pkgadd`.

Suggerimento: per installare Spatial Extender è preferibile utilizzare il wizard di installazione DB2 che dispone di un’interfaccia grafica di semplice utilizzo e di una guida per l’installazione, crea automaticamente utente e gruppi, e provvede alla configurazione dei protocolli e alla creazione dell’istanza. Se si decide di non utilizzare il wizard, è possibile installare Spatial Extender utilizzando lo script `db2_install` o il comando Solaris `pkgadd`. Si consiglia di utilizzare il comando Solaris `pkgadd` agli utenti esperti, solo se è necessario eseguire diverse operazioni di impostazione manuali durante il processo di installazione.

DB2 Spatial Extender è costituito da differenti funzioni e componenti che in ambiente Solaris vengono definiti pacchetti. Durante l’installazione di Spatial Extender mediante il comando `pkgadd`, è necessario installare tutti i pacchetti obbligatori e tutti i pacchetti associati alle funzioni facoltative che si desidera utilizzare. Il file `ComponentList.htm` contenuto nel CD di DB2 Spatial Extender, contiene l’elenco completo dei pacchetti che è necessario installare per DB2 Spatial Extender. Il file `ComponentList.htm` si trova nella directory `/cdrom/db2/solaris` dove `/cdrom` è il punto di mount del CD-ROM DB2 Spatial Extender.

Prerequisiti:

Prima di installare DB2 Spatial Extender per gli ambienti operativi Solaris:

- Assicurarsi che il sistema disponga dell’hardware e del software richiesti e di memoria sufficiente.
- Se si desidera eseguire l’installazione su un server o in un ambiente autonomo, è necessario che il prodotto DB2 Versione 8 server sia già installato.

Nota: Verificare che DB2 Spatial Client sia già installato. Il client DB2 Spatial Extender e i componenti di esempio sono integrati nelle installazioni del client e del server DB2. E’ possibile installare i componenti spaziali mediante l’installazione personalizzata di DB2, selezionando la funzione **Spatial Extender Client** in **Supporto client**, e la funzione **Esempi Spatial Extender** in **Strumenti di sviluppo applicazioni**. Se si desiderano solo le funzionalità del client spaziale e i componenti spaziali sono già stati installati con DB2, non sarà necessario eseguire la seguente procedura di installazione di DB2 Spatial Extender.

Introduzione

- Aggiornare i parametri di configurazione e riavviare il sistema per tutti i client e i server nell'ambiente operativo Solaris.
- E' necessario disporre di autorizzazione root.

Procedura:

Per installare DB2 Spatial Extender per Solaris Operating Environment utilizzando il wizard di installazione DB2, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD di DB2 Spatial Extender. Verrà aperta l'utilità Launchpad di installazione DB2, un'interfaccia da cui è possibile installare DB2 Spatial Extender. Per informazioni sull'attivazione di un CD, fare riferimento alla pubblicazione *DB2 for UNIX Quick Beginnings*.
3. Selezionare **Spatial Extender** e scegliere **AVANTI**.
4. Viene aperto il wizard di installazione DB2. Attenersi alle istruzioni indicate nel wizard per completare l'installazione. Se durante l'installazione si desiderano informazioni, è possibile avviare in qualsiasi momento la guida in linea scegliendo ?.

Per installare DB2 Spatial Extender utilizzando lo script db2_install, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD appropriato.
3. Per avviare lo script, immettere il comando **./db2_install**. Lo script db2_install si trova nella directory root nel CD dei prodotti di DB2 Versione 8. Lo script db2_install richiederà di inserire la parola chiave del prodotto.
4. Per installare DB2 Spatial Extender, immettere **DB2.GSE**.

Per installare DB2 Spatial Extender per Solaris mediante il comando **pkgadd**, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD di DB2 Spatial Extender.
3. Identificare i pacchetti obbligatori e i pacchetti facoltativi che si desidera installare. Il file ComponentList.htm contenuto nel CD di Spatial Extender, contiene l'elenco completo dei componenti che è necessario installare per DB2 Spatial Extender.
4. Eseguire il comando **pkgadd** per ciascun pacchetto che si desidera installare immettendo:

```
pkgadd nome_pacchetto
```

In questo comando *nome_pacchetto* è il pacchetto che si desidera installare.

Ad esempio, per installare Spatial Extender Base Server Support, è necessario installare il pacchetto db2gssg81 immettendo il seguente comando:

```
pkgadd db2gssg81
```

Al termine dell'installazione, il software Spatial Extender risulta installato nella directory `/opt/IBM/db2/V8.1`.

A questo punto creare l'istanza DB2 e verificare l'esito dell'installazione.

Argomenti correlati:

- "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26

Attività correlate:

- “Installing a DB2 product using pkgadd (Solaris Operating Environments)” nella *Installation and Configuration Supplement*
- “Installing a DB2 product using the db2_install script (UNIX)” nella *Installation and Configuration Supplement*
- “Mounting the CD-ROM (Solaris Operating Environment)” nella *Quick Beginnings for DB2 Servers*
- “Verifica dell’installazione di Spatial Extender” a pagina 39
- “Risoluzione dei problemi di installazione” a pagina 40

Installazione di DB2 Spatial Extender per Linux

E’ possibile installare DB2 Spatial Extender per Linux mediante il wizard di installazione DB2, lo script db2_install o il comando **rpm**.

Suggerimento: per installare Spatial Extender è preferibile utilizzare il wizard di installazione DB2 che dispone di un’interfaccia grafica di semplice utilizzo e di una guida per l’installazione, crea automaticamente utenti e gruppi, e provvede alla configurazione dei protocolli e alla creazione dell’istanza. Se si decide di non utilizzare il wizard, è possibile installare Spatial Extender utilizzando lo script db2_install o il comando **rpm**. Si consiglia di utilizzare il comando Linux **rpm** agli utenti esperti, solo se è necessario eseguire diverse operazioni di impostazione manuali durante il processo di installazione.

Prerequisiti:

Prima di installare DB2 Spatial Extender per Linux, procedere come segue:

- Assicurarsi che il sistema disponga dell’hardware e del software richiesti e di memoria sufficiente.
- Se si desidera eseguire l’installazione su un server o in un ambiente autonomo, assicurarsi che DB2 Versione 8 sia già installato.

Nota:

- Verificare che DB2 Spatial Client sia già installato. Il client DB2 Spatial Extender e i componenti di esempio sono inclusi in DB2 client e server. E’ possibile installare i componenti spaziali mediante l’installazione personalizzata di DB2, selezionando la funzione **Spatial Extender Client in Supporto client** e la funzione **Esempi Spatial Extender in Strumenti di sviluppo applicazioni**. Se si desiderano solo le funzionalità del client spaziale e i componenti spaziali sono già stati installati con DB2, non sarà necessario eseguire la seguente procedura di installazione di DB2 Spatial Extender.
- Aggiornare i parametri di configurazione e riavviare il sistema per tutti i client e i server DB2 con sistema operativo Linux.
- E’ necessario disporre di autorizzazione root.

Procedura:

Per installare DB2 Spatial Extender utilizzando il wizard di installazione DB2, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Inserire ed eseguire il mount del CD di DB2 Spatial Extender. Verrà aperta l’utilità Launchpad di installazione DB2, un’interfaccia da cui è possibile

installare DB2 Spatial Extender. Per informazioni sul mount del CD, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.

3. Scegliere **Installa prodotti**.
4. Selezionare **Spatial Extender** e scegliere **AVANTI**.
5. Selezionare l'opzione desiderata nel wizard di installazione DB2. E' possibile installare **DB2 Spatial Extender** o il **DB2 Application Development Client**.
 - Se si desidera utilizzare solo le funzionalità del client spaziale, scegliere **DB2 Application Development Client** e selezionare le seguenti funzioni:
 - **Client Spatial Extender in Supporto client**
 - Facoltativo: **Esempi Spatial Extender in Strumenti di sviluppo applicazioni**
 - Se si desidera utilizzare le funzionalità del server e del client spaziali, scegliere **DB2 Spatial Extender** e selezionare le seguenti funzioni:
 - **Supporto server di base Spatial Extender in Supporto server**
 - **Client Spatial Extender in Supporto client Spatial Extender**
 - Facoltativo: **Esempi Spatial Extender in Strumenti di sviluppo applicazioni**

Attenersi alle istruzioni indicate nel wizard per completare l'installazione. Se durante l'installazione si desiderano informazioni, è possibile avviare in qualsiasi momento la guida in linea scegliendo ?.

Per installare DB2 Spatial Extender utilizzando lo script `db2_install`, procedere come segue:

1. Collegarsi come utente con autorizzazione `root`.
2. Inserire ed eseguire il mount del CD appropriato.
3. Per avviare lo script, immettere il comando `./db2_install`. Lo script `db2_install` si trova nella directory `root` nel CD dei prodotti di DB2 Versione 8. Lo script `db2_install` richiederà di inserire la parola chiave del prodotto.
4. Per installare DB2 Spatial Extender, immettere **DB2.GSE**.

Per installare Spatial Extender per Linux mediante il comando `rpm`, procedere come segue:

1. Collegarsi come utente con autorizzazione `root`.
2. Abilitare il sistema per l'installazione di DB2 per Linux. Per ulteriori informazioni, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.
3. Inserire ed eseguire il mount del CD di DB2 Spatial Extender.
4. Identificare i pacchetti obbligatori e i pacchetti facoltativi che si desidera installare.

Nota: Il file `ComponentList.htm` contenuto nel CD di DB2 Spatial Extender, contiene l'elenco completo dei componenti che è necessario installare per DB2 Spatial Extender.

5. Eseguire il comando `rpm` per ciascun pacchetto che si desidera installare:
`rpm -ivh nome_pacchetto`

Ad esempio, se si desidera installare il server, installare il pacchetto `IBM_db2gssg81-8.1.0-0.i386.rpm` immettendo il seguente comando:

```
rpm -IBM_db2gssg81-8.1.0-0.i386.rpm
```

Al termine dell'installazione, Spatial Extender risulta installato nella directory /opt/IBM/db2/V8.1 insieme agli altri prodotti DB2.

A questo punto creare l'istanza DB2 e verificare l'esito dell'installazione.

Argomenti correlati:

- "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26

Attività correlate:

- "Installing a DB2 product using rpm (Linux)" nella *Installation and Configuration Supplement*
- "Mounting the CD-ROM (Linux)" nella *Quick Beginnings for DB2 Servers*
- "Installing a DB2 product using the db2_install script (UNIX)" nella *Installation and Configuration Supplement*
- "Verifica dell'installazione di Spatial Extender" a pagina 39
- "Risoluzione dei problemi di installazione" a pagina 40

Creazione dell'istanza di DB2 Spatial Extender

Questa attività fa parte del processo di impostazione di Spatial Extender.

La sezione è valida solo per piattaforme UNIX.

DB2 Spatial Extender può essere utilizzato con qualsiasi istanza DB2 creata dopo l'installazione del codice Spatial Extender.

Il comando **db2icrt** viene utilizzato per creare nuove istanze DB2. DB2 Spatial Extender viene incluso in tutte le istanze DB2 create dopo l'installazione del prodotto.

Le istanze DB2 create prima dell'installazione di Spatial Extender, invece, non contengono riferimenti al prodotto. Per aggiornare un'istanza DB2 esistente, utilizzare il comando **db2iupdt**. Se si dispone di un'istanza creata per il server di gestione DB2 mediante il Centro di controllo DB2, prima dell'installazione di DB2 Spatial Extender, sarà necessario aggiornarla.

Procedura:

Per aggiornare un'istanza utilizzando il comando **db2iupdt**, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Eseguire il comando:

```
DB2DIR/instance/db2iupdt -a TipoAut -u IDprotetto NomeIst
```

Dove:

DB2DIR

La directory di installazione DB2.

- In ambiente AIX, la directory di installazione è /usr/opt/db2_08_01
- In tutti gli altri sistemi operativi basati su UNIX, la directory di installazione è /opt/IBM/db2/V8.1

-a Tipoaut

rappresenta il tipo di autenticazione dell'istanza che può essere uno tra

SERVER, CLIENT, DCS, SERVER_ENCRYPT, DCS_ENCRYPT. SERVER è l'impostazione predefinita. Il parametro è facoltativo.

-u IDprotetto

rappresenta il nome dell'utente per il quale verranno eseguite le UDF (user defined function) e le procedure memorizzate protette. Se l'istanza viene creata su un client DB2, questa indicazione non è obbligatoria. Indicare il nome dell'utente protetto creato.

NomeIst

Rappresenta il nome dell'istanza che deve corrispondere al nome dell'istanza dell'utente proprietario. Indicare il nome dell'istanza creato. L'istanza verrà creata nella directory principale dell'utente proprietario dell'istanza.

Per creare un'istanza utilizzando il comando **db2icrt**, procedere come segue:

1. Collegarsi come utente con autorizzazione root.
2. Eseguire il comando:

```
DB2DIR/instance/db2icrt -a TipoAut -u IDprotetto NomeIst
```

Dove:

DB2DIR

corrisponde alla directory di installazione del DB2.

- In ambiente AIX, la directory di installazione è /usr/opt/db2_08_01
- In tutti gli altri sistemi operativi basati su UNIX, la directory di installazione è /opt/IBM/db2/V8.1

-a Tipoaut

rappresenta il tipo di autenticazione dell'istanza che può essere uno tra SERVER, CLIENT, DCS, SERVER_ENCRYPT, DCS_ENCRYPT. SERVER è l'impostazione predefinita. Il parametro è facoltativo.

-u IDprotetto

rappresenta il nome dell'utente per il quale verranno eseguite le UDF (user defined function) e le procedure memorizzate protette. Se l'istanza viene creata su un client DB2, questa indicazione non è obbligatoria. Indicare il nome dell'utente protetto creato.

NomeIst

Rappresenta il nome dell'istanza che deve corrispondere al nome dell'istanza dell'utente proprietario. Indicare il nome dell'istanza creato. L'istanza verrà creata nella directory principale dell'utente proprietario dell'istanza.

Ad esempio, se viene utilizzata l'autenticazione server, l'utente protetto sarà db2fenc1 e l'istanza a cui appartiene l'utente db2inst1. Per creare un'istanza su un sistema AIX, Utilizzare il seguente comando:

```
/usr/opt/db2_08_01/instance/db2icrt -a server -u db2fenc1 db2inst1
```

Dopo aver creato un'istanza è possibile configurare la funzione di notifica per il controllo sullo stato dell'istanza mediante il Centro di controllo stato o il processore riga comandi. Per ulteriori informazioni, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.

Verifica dell'installazione di Spatial Extender

L'impostazione e la configurazione di Spatial Extender prevede una fase di verifica dell'installazione.

Per verificare che DB2 Spatial Extender sia stato installato e configurato in modo corretto, creare un database ed eseguire il programma di controllo dell'installazione.

E' possibile verificare l'installazione utilizzando il programma di esempio di DB2 Spatial Extender, runGseDemo. Il programma runGseDemo è stato ideato per evidenziare eventuali problemi di installazione. Durante la verifica dell'installazione, è possibile che vengano visualizzati messaggi di errore che aiutano ad individuare problemi di sistema specifici. La maggior parte dei messaggi di errore sono causati da un piccolo numero di problemi noti. Per evitare questi errori, fare riferimento al paragrafo "Prerequisiti".

Le seguenti operazioni per la verifica dell'installazione riguardano i seguenti sistemi operativi: Windows, AIX, HP-UX, Solaris Operating Environments, Linux per Intel e Linux per S/390.

Prerequisiti:

Prima di eseguire il programma runGseDemo:

- Assicurarsi di aver installato DB2 Spatial Extender negli ambienti appropriati.
- Utilizzare un nuovo database a cui non è stata associata alcuna operazione spaziale.
- Per le installazioni su UNIX (AIX, HP-UX, Solaris Operating Environments, Linux per Intel e Linux per S/390), assicurarsi che l'istanza di DB2 Spatial Extender sia stata creata. Per ulteriori informazioni sull'esecuzione del programma **db2ilist** per la verifica delle istanze, consultare la pubblicazione *DB2 Installation and Configuration Supplement*. Per avviare l'istanza DB2, potrebbe essere necessario dover eseguire il comando **db2start**.
- Aumentare il valore del parametro di configurazione del database relativo alla dimensione heap. Per ulteriori informazioni, consultare la sezione relativa alla risoluzione dei problemi che si sono verificati durante l'installazione.

Procedura:

Per verificare l'esito dell'installazione, procedere come segue:

1. Solo in ambiente UNIX, collegarsi come proprietario dell'istanza.
2. Creare un database.

Aprire la finestra Comandi DB2 ed immettere:

```
db2 create database miodb
```

dove *miodb* è il nome del database.

3. Individuare il programma di verifica dell'installazione.

- a. In sistemi operativi UNIX, immettere:

```
cd $HOME/sql/lib/samples/spatial
```

dove *\$HOME* è la directory principale del proprietario dell'istanza.

- b. In Windows, immettere:

```
cd c:\Programmi\IBM\sql\lib\samples\spatial
```

dove `c:\Programmi\IBM\sqllib` è la directory in cui è stato installato DB2 Spatial Extender.

4. Eseguire il programma di controllo dell'installazione.

Immettere **runGseDemo** nella riga comandi DB2, ad esempio:

```
runGseDemo miodb IDutente password
```

dove *miodb* è il nome del database.

Se durante il processo di verifica vengono visualizzati messaggi di errore, è necessario risolvere i problemi che si sono verificati durante l'installazione.

Attività correlate:

- "Risoluzione dei problemi di installazione" a pagina 40
- "Installazione di DB2 Spatial Extender per Windows" a pagina 27
- "Installazione di DB2 Spatial Extender per AIX" a pagina 29
- "Installazione di DB2 Spatial Extender per HP-UX" a pagina 31
- "Installazione di DB2 Spatial Extender per Solaris Operating Environment" a pagina 33
- "Installazione di DB2 Spatial Extender per Linux" a pagina 35

Risoluzione dei problemi di installazione

Quando viene eseguito il programma di esempio runGseDemo per verificare l'esito dell'installazione di Spatial Extender, è possibile che vengano rilevati i seguenti errori:

Database già abilitato per operazioni spaziali

Controllare che il database di cui si sta verificando l'installazione sia nuovo e non presenti operazioni spaziali ad esso associate; in caso contrario, il programma di esempio non funzionerà.

Se il database su cui si esegue il programma di esempio è già abilitato per operazioni spaziali, si riceverà il seguente messaggio di errore:

```
Enabling database logtst...  
Returning from ENABLE_DB:  
Return code = -14  
Return message text =  
GSE0014E Il database è già stato abilitato per operazioni spaziali.
```

Per correggere questo errore, eliminare il database e ripetere le operazioni descritte nella sezione Verifica dell'installazione di Spatial Extender.

Valore del parametro di configurazione di database manager relativo alla dimensione heap dell'applicazione

Se il valore del parametro APPLHEAPSZ non è corretto, durante l'abilitazione del database per l'esecuzione delle operazioni spaziali, viene ricevuto questo messaggio di errore:

```
GSE0213N Un'operazione bind non è riuscita.  
SQLERROR = "SQL0001N Bind o precompilazione  
non riusciti.  
SQLSTATE=00000".SQLSTATE=57011
```

Per aumentare il valore del parametro di configurazione del database relativo alla dimensione heap, immettere:

```
db2 update db cfg for nome_database APPLHEAPSZ 2048
```

Se il valore 2048 non è adeguato, aumentare il parametro APPLHEAPSZ con incrementi di 256.

Considerazioni successive all'installazione

Dopo aver installato Spatial Extender, procedere come segue:

- Scaricare ArcExplorer
- Accedere ai dati di riferimento del geocoder

Download di ArcExplorer per DB2

IBM fornisce un browser, prodotto dalla ESRI (Environmental Systems Research Institute) per IBM, in grado di produrre direttamente risultati visivi delle interrogazioni dei dati di DB2 Spatial Extender senza utilizzare un server di dati intermedio. Si tratta del ArcExplorer for DB2. E' possibile scaricare ArcExplorer for DB2 gratuitamente al sito Web di IBM Spatial Extender, al seguente indirizzo:

<http://www.ibm.com/software/data/spatial/>

Per ulteriori informazioni sull'installazione e l'utilizzo di ArcExplorer for DB2, consultare la pubblicazione *Using ArcExplorer*, disponibile come parte del download di ArcExplorer for DB2 al sito Web di DB2 Spatial Extender.

Importante: DB2 Universal Database Versione 8.1 viene fornito con IBM Software Developer's Kit (SDK) per Java Versione 1.3.1. Quando si installa ArcExplorer for DB2, collocarlo in una directory diversa da quella in cui si trova DB2. Impostare la variabile di ambiente CLASSPATH.

Accesso ai dati di riferimento del geocoder

I dati presenti sul CD Dati di riferimento del geocoder di DB2 Spatial Extender sono stati creati per essere utilizzati con geocoder forniti da Spatial Extender. Si tratta di dati ottenuti dalle carte geografiche della rete stradale negli Stati Uniti. Il geocoder DB2SE_USA utilizza questi dati per determinare le coordinate di indirizzi in un database abilitato per operazioni spaziali. I dati della carta geografica di base vengono definiti *dati di riferimento*. Il geocoder DB2SE_USA carica i dati relativi gli indirizzi (non spaziali) nel database, li confronta e li mette in corrispondenza con i dati di riferimento, infine li converte in coordinate che possono essere memorizzate da DB2 Spatial Extender. Questo processo viene definito *geocoding*.

I dati di riferimento forniti nel CD includono il file EDGELocator.loc, utilizzato dal geocoder DB2SE_USA per individuare determinati dati di riferimento. Ad esempio, se si effettua il geocoding di indirizzi della California, del Kentucky e dell'Oregon, il geocoder DB2SE_USA utilizzerà il file indicatore di posizione presente sul CD per determinare la posizione degli indirizzi.

Procedura:

È possibile accedere ai dati del geocoder direttamente dal CD oppure è possibile copiare i dati sul disco fisso. Per copiare i file dei dati del geocoder dal CD al proprio ambiente server DB2 Spatial Extender, procedere come segue.

In sistemi operativi UNIX:

Introduzione

1. Eseguire il mount del CD. Per informazioni sull'attivazione di un CD, fare riferimento alla pubblicazione *DB2 for UNIX Quick Beginnings*.
 2. Collegarsi alla macchina server di destinazione come utente con autorizzazione root.
 3. Immettere uno dei seguenti argomenti:
 - Per AIX:

```
cp /cdrom/db2/* /usr/opt/db2_08_01/gse/refdata/
```
 - Per le altre piattaforme UNIX:

```
cp /cdrom/db2/* /opt/IBM/db2/V8.1/gse/refdata/
```
- Nota:** E' possibile copiare tutti i file di dati del geocoder in qualsiasi directory dell'unità locale. Una volta stabilita la directory in cui copiare i file, sarà necessario modificare il file indicatore di posizione in modo che faccia riferimento alla nuova ubicazione.
4. Scollegarsi.

In ambiente Windows, è possibile utilizzare una finestra di comandi o Windows Explorer.

Per accedere ai dati del geocoder attraverso la finestra Comandi, procedere come segue:

1. Scegliere **Start** -> **Programmi** -> **IBM DB2** -> **Finestra Comandi**.
2. Immettere il seguente comando:

```
copy d:\db2\* %db2path%\gse\refdata
```

dove *d*: è la lettera che corrisponde all'unità CD.

Nota: E' possibile copiare tutti i file di dati del geocoder in qualsiasi directory dell'unità locale. Una volta stabilita la directory in cui copiare i file, sarà necessario modificare il file indicatore di posizione in modo che faccia riferimento alla nuova ubicazione.

Per accedere ai dati del geocoder attraverso Windows Explorer, procedere come segue:

Copiare tutti i file da *d:\db2* in *c:\Programmi\IBM\sqliib\gse\refdata*, dove *d*: rappresenta l'unità CD e *c:\Programmi\IBM\sqliib* è la directory in cui è stato installato DB2.

Attività correlate:

- "Impostazione e installazione di Spatial Extender" a pagina 25

CD per dati e carte geografiche di DB2 Spatial Extender

DB2 Spatial Extender viene fornito con sette CD contenenti dati e carte geografiche.

Le informazioni relative ai dati e alle carte geografiche, indicate come Dati e carte geografiche DB2 Spatial Extender 1 - 7, sono contenute in sette CD. La seguente tabella fornisce un riepilogo dei dati presenti su ciascun CD.

Tabella 2. Informazioni sul CD di dati e carte geografiche

CD Dati e carte geografiche	Riepilogo del tipo di dati delle carte geografiche
CD 1	Europa e resto del mondo
CD 2	Canada, Messico e Stati Uniti
CD 3	Stati Uniti
CD 4	Stati Uniti (regione occidentale)
CD 5	Stati Uniti (regione centrale)
CD 6	Stati Uniti (regione orientale)
CD 7	Stati Uniti (regione meridionale)

Per una descrizione dettagliata dei dati forniti da ESRI, consultare il file di aiuto ESRI, *esridata.hlp*, situato sul CD Dati e carte geografiche di DB2 Spatial Extender.

- In ambiente Windows, visualizzare il file di aiuto in *x: esridata.hlp*, dove *x* corrisponde all'unità CD.
- In sistemi operativi UNIX, visualizzare o stampare il file di aiuto contenuto nel CD in */cdrom/esridata.hlp*, dove */cdrom* indica il punto di mount.

Attività correlate:

- "Impostazione ed abilitazione di DB2 Geodetic Extender" a pagina 165

Capitolo 5. Migrazione dell'ambiente Spatial Extender a DB2 Universal Database Versione 8

Questa sezione illustra come migrare DB2 Spatial Extender dalla Versione 8.1 alla Versione 8.2. Inoltre descrive come utilizzare il programma di utilità per eseguire la migrazione da un ambiente a 32-bit a un ambiente 64-bit.

Migrazione di un database abilitato per le funzioni spaziali

Se è stato utilizzato DB2 Spatial Extender Versione 8.1, sarà necessario attenersi alle istruzioni riportate di seguito prima di utilizzare un database abilitato per le funzioni spaziali con DB2 Spatial Extender Versione 8.2 o DB2 Geodetic Extender Versione 8.2. Questa sezione descrive le operazioni da eseguire per migrare i database abilitati per le funzioni spaziali da una versione precedente di DB2 Spatial Extender.

Prerequisiti:

Prima di avviare il processo di migrazione:

- Prima di eseguire l'utilità per la migrazione, chiudere tutte le connessioni al database.
- Assicurarsi che il sistema disponga di tutti i requisiti di installazione per DB2 Spatial Extender Versione 8.2.
- Per creare la copia di riserva di un database, è necessario di disporre dell'autorizzazione di SYSADM, SYSCTRL, o SYSMAINT.
- Per migrare un database, è necessaria l'autorizzazione SYSADM.

Procedura:

Per migrare l'ambiente DB2 Spatial Extender, procedere come segue:

1. Creare una copia di riserva dei database creati con la versione 8.1. Per informazioni sulla creazione di copie di riserva dei database, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.
2. Installare DB2 Universal Database Versione 8.2 e DB2 Spatial Extender Versione 8.2.
3. Migrare l'istanza e i database DB2 dalla Versione 8.1 alla Versione 8.2. Per informazioni sulla migrazione dell'istanza e dei database DB2, consultare la pubblicazione *DB2 Installation and Configuration Supplement*.
4. Migrare un database abilitato per le funzioni spaziali dalla Versione 8.1 alla Versione 8.2 utilizzando l'utilità per la migrazione di Spatial Extender.
 - a. Dal prompt dei comandi del sistema operativo, eseguire il comando **db2se migrate_v82** per migrare il database.

```
db2se migrate_v82 database_name userId user_id PW password
```

Per la sintassi di questo comando, consultare la sezione "Comando db2se migrate_v82" a pagina 47.

Messaggi di migrazione

Se la migrazione ha esito positivo, viene visualizzato un messaggio simile al seguente:

```
GSE0000I Operazione eseguita correttamente
```

In caso contrario viene visualizzato un messaggio simile al seguente:

```
GSE9002N Si è verificato un errore durante l'esecuzione della migrazione del database Spatial Extender.
```

Nota: Durante la migrazione potrebbero verificarsi i seguenti errori:

- Il database non è abilitato per le funzioni spaziali
- Il database non è versione 8.1 e abilitato per le funzioni spaziali.
- Il database è già versione 8.2 e abilitato per le funzioni spaziali.
- Il nome del database non è valido.
- Esistono altre connessioni al database e non può essere eseguito.
- Il catalogo spaziale non è coerente
- L'utente non è autorizzato.
- La password non è valida.
- Non è possibile migrare alcuni oggetti utente.

Per qualsiasi errore ricevuto, fare riferimento al file dei messaggi, che contiene anche altre informazioni, quali gli indici, le viste e l'impostazione di geocoding che sono stati migrati.

Attività correlate:

- "Backing up databases before DB2 migration" nella *Quick Beginnings for DB2 Servers*
- "Migrating databases" nella *Quick Beginnings for DB2 Servers*
- "Migrating instances (UNIX)" nella *Quick Beginnings for DB2 Servers*
- "Migrating DB2 UDB (Windows)" nella *Quick Beginnings for DB2 Servers*
- "Migrating DB2 UDB (UNIX)" nella *Quick Beginnings for DB2 Servers*
- "Migrating DB2 Personal Edition (Windows)" nella *Quick Beginnings for DB2 Personal Edition*
- "Migrating DB2 Personal Edition (Linux)" nella *Quick Beginnings for DB2 Personal Edition*
- "Migrating databases on DB2 Personal Edition (Windows)" nella *Quick Beginnings for DB2 Personal Edition*
- "Migrating instances and databases on DB2 Personal Edition (Linux)" nella *Quick Beginnings for DB2 Personal Edition*

Riferimenti correlati:

- "Comando db2se migrate_v82" a pagina 47

Capitolo 6. Impostazione di un database

Questo capitolo spiega come configurare un database in modo che possa ricevere i dati spaziali.

Configurazione di un database per l'inserimento dei dati spaziali

DB2 Spatial Extender, eseguito in ambiente DB2 Universal Database, funziona con la maggior parte dei valori predefiniti di DB2. Tuttavia, le operazioni spaziali sono influenzate da diversi parametri di configurazione. Affinché le applicazioni spaziali vengano eseguite correttamente, è necessario ottimizzare tali parametri. In alcuni casi, durante l'esecuzione di un'operazione spaziale, è obbligatorio scegliere un valore diverso da quello predefinito. In altri casi, tale scelta è solo consigliata, e la selezione del valore dipende dalle applicazioni e dall'ambiente DB2 utilizzati. In questa sezione sono indicati i parametri di configurazione di DB2 che interferiscono con le operazioni di DB2 Spatial Extender.

La sezione seguente descrive come impostare il database manager DB2 e i parametri di configurazione che interferiscono con le operazioni di DB2 Spatial Extender.

Ottimizzazione dei parametri di configurazione del database

Diversi parametri di configurazione del database influenzano le applicazioni spaziali. Per modificare i parametri di configurazione del database, è necessario essere collegati al database. Dopo aver modificato i valori dei parametri di un database, le modifiche avranno effetto solo in quel database. Le sezioni seguenti indicano come ottimizzare i parametri delle applicazioni spaziali:

- "Ottimizzazione delle caratteristiche della registrazione della transazione"
- "Ottimizzazione della dimensione heap dell'applicazione" a pagina 51
- "Ottimizzazione della dimensione heap di controllo dell'applicazione" a pagina 51

Ottimizzazione delle caratteristiche della registrazione della transazione

Prima di abilitare un database per le operazioni spaziali, verificare che la capacità di registrazione per la transazione sia sufficiente. I valori predefiniti dei parametri di configurazione per la registrazione della transazione non assicurano una capacità sufficiente per lo svolgimento dell'operazione, se si intende:

- Abilitare un database per le operazioni spaziali in ambiente Windows
- Importare dai file shape utilizzando la procedura memorizzata ST_import_shape
- Utilizzare la funzione di geocoding con un intervallo di commit elevato
- Eseguire più transazioni contemporaneamente

Se si desidera eseguire una delle attività descritte, è necessario aumentare la capacità di registrazione della transazione del database modificando uno o più parametri di configurazione. In caso contrario, è possibile utilizzare le impostazioni predefinite. Per utilizzare le impostazioni predefinite, passare alla sezione "Ottimizzazione della dimensione heap dell'applicazione" a pagina 51.

Impostazione di un database

Suggerimento: fare riferimento ai valori minimi consigliati nella seguente tabella, relativi ai tre parametri di configurazione per la registrazione della transazione.

Tabella 3. Valori minimi consigliati per i parametri di configurazione della transazione

Parametro	Descrizione	Valore predefinito	Valore minimo consigliato
LOGFILSZ	Indica la dimensione del file di registrazione come un numero costituito da blocchi di 4 KB	1000	1000
LOGPRIMARY	Indica quanto file di registrazione principali assegnare come file di registrazione di ripristino	3	10
LOGSECOND	Indica il numero di file di registrazione secondari	2	2

Se la capacità del file di registrazione della transazione non è sufficiente, quando si cercherà di abilitare un database per le operazioni spaziali, verrà visualizzato il seguente messaggio di errore:

GSE0010N Spazio di registrazione disponibile non sufficiente per DB2.

Procedura:

Per aumentare il valore di uno o più parametri di configurazione, procedere come segue:

1. Individuare il valore corrente dei parametri LOGFILSZ, LOGPRIMARY e LOGSECOND visualizzando l'output con il comando GET DATABASE CONFIGURATION o dalla finestra di **configurazione del database** del Centro di controllo DB2.
2. Stabilire se modificare uno, due o tre valori come indicato nella tabella seguente.
3. Modificare i valori che si desidera modificare. E' possibile modificare i valori utilizzando i seguenti comandi, in cui *nome_db* indica il database utilizzato:

```
UPDATE DATABASE CONFIGURATION FOR nome_db USING LOGFILSZ 1000
```

```
UPDATE DATABASE CONFIGURATION FOR nome_db USING LOGPRIMARY 10
```

```
UPDATE DATABASE CONFIGURATION FOR nome_db USING LOGSECOND 2
```

Se viene modificato solo il parametro LOGSECOND, la modifica ha effetto immediatamente. Per utilizzare le impostazioni predefinite, passare alla sezione "Ottimizzazione della dimensione heap dell'applicazione" a pagina 51.

4. Se viene modificato il parametro LOGFILSZ o LOGPRIMARY o entrambi:
 - a. Disconnettere tutte le applicazioni dal database.
 - b. Se il database era stato esplicitamente attivato, disattivarlo.

Affinché le modifiche ai parametri LOGFILSZ e LOGPRIMARY abbiano effetto sarà necessario attivare o connettersi nuovamente al database.

Ottimizzazione della dimensione heap dell'applicazione

Per indicare la dimensione heap dell'applicazione (in numeri pagine di 4 KB), è possibile utilizzare il parametro APPLHEAPSZ. Questo parametro definisce il numero di pagine di memoria private utilizzabili da database manager per conto di un determinato agente o subagente. Il valore heap viene assegnato al momento dell'inizializzazione di un agente o subagente per un'applicazione. Viene assegnata una quantità minima necessaria per elaborare la richiesta dell'agente o del subagente. Se l'agente o il subagente richiede più spazio heap per elaborare istruzioni SQL di grandi dimensioni, database manager assegna la memoria necessaria fino al valore massimo indicato da questo parametro. L'heap dell'applicazione viene assegnato esternamente alla memoria privata dell'agente.

Il valore predefinito per il parametro APPLHEAPSZ è 128 (pagine di 4 KB). Se si esegue la procedura memorizzata ST_enable_db, tale valore dovrà essere almeno 2048.

Suggerimento: Per la maggior parte di applicazioni DB2 Spatial Extender, soprattutto quelle importate o esportate dai file shape, è preferibile che il valore del parametro APPLHEAPSZ sia almeno 2048.

Se il valore di APPLHEAPSZ non è corretto, quando si cercherà di abilitare un database per le operazioni spaziali, verrà visualizzato il seguente messaggio di errore:

```
GSE0009N Spazio non sufficiente nell'heap dell'applicazione DB2.
```

```
GSE0213N Operazione bind non riuscita. SQLERROR = "SQL0001N Bind o precompilazione non completato correttamente. SQLSTATE=00000".
```

Procedura:

Per modificare la dimensione heap dell'applicazione, procedere come segue:

1. Individuare il valore corrente del parametro APPLHEAPSZ visualizzando l'output con il comando GET DATABASE CONFIGURATION o dalla finestra di **configurazione database** del Centro di controllo DB2.
2. Modificare il valore in 2048 o in un valore superiore. E' possibile modificare il valore utilizzando il seguente comando in cui *nome_db* indica il database utilizzato:


```
UPDATE DATABASE CONFIGURATION FOR nome_db USING APPLHEAPSZ 2048
```
3. Disconnettere tutte le applicazioni dal database.
4. Se il database era stato esplicitamente attivato, disattivarlo.

Affinché le modifiche abbiano effetto sarà necessario attivare o connettersi nuovamente al database.

Ottimizzazione della dimensione heap di controllo dell'applicazione

Per la maggior parte di applicazioni DB2 Spatial Extender, soprattutto quelle importate o esportate dai file shape, è preferibile che il valore della dimensione heap di controllo dell'applicazione sia quello consigliato. Tale impostazione viene effettuata mediante il parametro APP_CTL_HEAP_SZ, che indica la dimensione massima, in pagine di 4 KB, della memoria condivisa del controllo dell'applicazione. Gli heap di controllo dell'applicazione vengono assegnati in base a questa memoria condivisa. Per ciascun database in cui l'applicazione è attiva, viene assegnato un heap di controllo dell'applicazione (oppure, nel caso di un

Impostazione di un database

sistema con database suddivisi in partizioni, per ciascuna partizione in cui l'applicazione è attiva). L'heap viene assegnato durante l'elaborazione della connessione eseguita dal primo agente che riceve una richiesta per l'applicazione nel database (o nella partizione del database). L'heap viene utilizzato per condividere le informazioni tra gli agenti che lavorano per conto della stessa applicazione. In un ambiente costituito da database suddivisi in partizioni, la condivisione avviene a livello delle partizioni del database e non tra le partizioni stesse. Il valore predefinito per il parametro APP_CTL_HEAP_SZ è 128.

Suggerimento: Per la maggior parte delle applicazioni di DB2 Spatial Extender, è preferibile che il valore del parametro APP_CTL_HEAP_SZ sia almeno 1024 (pagine di 4 KB).

Se il valore di APP_CTL_HEAP_SZ non è corretto, quando si cercherà di importare i dati in un database da un file shape, verrà visualizzato il seguente messaggio di errore:

```
GSE0214N Istruzione INSERT non riuscita. SQLERROR = "SQL0973N Spazio di memorizzazione disponibile non sufficiente nell'heap "APP_CTL_HEAP" per elaborare l'istruzione.
```

Procedura:

Per modificare la dimensione heap di controllo dell'applicazione, procedere come segue:

1. Individuare il valore corrente del parametro APP_CTL_HEAP_SZ visualizzando l'output con il comando GET DATABASE CONFIGURATION o dalla finestra di **configurazione database** del Centro di controllo DB2.
2. Modificare il valore in 1024 (pagine di 4 KB) o in un valore superiore. E' possibile utilizzare il seguente comando in cui *nome_db* indica il database utilizzato:

```
UPDATE DATABASE CONFIGURATION FOR nome_db USING APP_CTL_HEAP_SZ 1024
```
3. Disconnettere tutte le applicazioni dal database.
4. Se il database era stato esplicitamente attivato, disattivarlo.

Affinché le modifiche abbiano effetto sarà necessario attivare o connettersi nuovamente al database.

Capitolo 7. Impostazione delle risorse spaziali per un database

Dopo aver impostato il database affinché possa ricevere i dati spaziali, fornire al database le risorse necessarie per la creazione e la gestione delle colonne spaziali e l'analisi dei dati spaziali. Tali risorse comprendono:

- Oggetti forniti da Spatial Extender per supportare le operazioni spaziali; ad esempio, le procedure memorizzate per la gestione dei database, i tipi di dati spaziali, la funzione di geocoding e l'utilità per l'importazione e l'esportazione dei dati.
- Dati di riferimento: indirizzi utilizzati da DB2SE_USA_GEOCODER per convertire singoli indirizzi in coordinate.
- Qualsiasi geocoder fornito dagli utenti o dai fornitori.

Questo capitolo descrive queste risorse ed introduce le attività da svolgere affinché sia possibile utilizzarle: l'abilitazione del database per le operazioni spaziali, l'impostazione dell'accesso ai dati di riferimento e la registrazione dei geocoder non predefiniti.

Impostazione delle risorse nel database

La prima attività da eseguire dopo l'impostazione del database affinché possa ricevere i dati spaziali, consiste nell'abilitare il database per le operazioni spaziali, quali l'inserimento dei dati spaziali nelle tabelle e l'elaborazione delle interrogazioni spaziali. L'attività consiste nel caricare il database con determinate risorse fornite da DB2 Spatial Extender. Le risorse e l'attività sono descritte nella seguente sezione.

Inventario delle risorse per il database

Affinché un database supporti l'esecuzione delle funzioni spaziali, DB2[®] Spatial Extender fornisce al database le seguenti risorse:

- Procedure memorizzate. Se viene effettuata un'operazione spaziale, ad esempio viene emesso un comando per importare dati spaziali, per eseguire l'azione richiesta DB2 Spatial Extender richiama una procedura memorizzata.
- Tipi di dati spaziali. E' necessario assegnare un tipo di dati spaziali a ciascuna colonna della vista o della tabella in cui si desidera memorizzare i dati spaziali.
- Catalogo di DB2 Spatial Extender. Diverse operazioni si basano sul catalogo. Ad esempio, per accedere ad una colonna spaziale mediante gli strumenti di visualizzazione, è possibile che la colonna spaziale debba essere registrata nel catalogo.
- Un indice di griglia spaziale. Consente di definire gli indici di griglia nelle colonne spaziali.
- Funzioni spaziali. E' possibile utilizzarle per gestire i dati spaziali; ad esempio per determinare i rapporti tra le forme geometriche e per generare più dati spaziali.
- Definizioni di sistemi di coordinate.
- Sistemi di riferimento spaziali predefiniti.

Impostazione delle risorse spaziali per un database

- Due schemi: DB2GSE e ST_INFORMTN_SCHEMA. DB2GSE contiene gli oggetti appena elencati: procedure memorizzate, tipi di dati spaziali, il catalogo DB2 Spatial Extender, e così via. Le viste del catalogo sono contenute anche in ST_INFORMTN_SCHEMA per conformità agli standard SQL/MM.

Abilitazione di un database per le operazioni spaziali

Con la definizione “abilitazione del database per le operazioni spaziali”, generalmente si intende il processo mediante il quale si ottiene da DB2 Spatial Extender un database completo di risorse che consentono di creare colonne spaziali e gestire dati spaziali.

Prerequisiti:

Per abilitare un database per le operazioni spaziali, è necessario che l’ID utente disponga dell’autorizzazione SYSADM o DBADM.

Limitazioni:

E’ possibile utilizzare solo i tipi di dati creati mediante il comando `enable_db`.

Procedura:

E’ possibile abilitare un database per le operazioni spaziali nei seguenti modi:

- Utilizzando la finestra **Abilita database** dal menu **DB2 Spatial Extender**. Il menu è disponibile nell’oggetto database del Centro di controllo DB2.
- Immettendo il comando **db2se enable_db**.
- Eseguendo un’applicazione che richiami la procedura memorizzata `db2gse.ST_enable_db`.

Nota:

E’ possibile indicare il tablespace in cui si desidera inserire il catalogo di DB2 Spatial Extender. In caso contrario, DB2 utilizzerà il tablespace predefinito.

Argomenti correlati:

- “Inventario delle risorse per il database” a pagina 53

Attività correlate:

- “Scrittura di applicazioni per DB2 Spatial Extender” a pagina 133

Riferimenti correlati:

- “Richiamo dei comandi per l’impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125
- “ST_enable_db” a pagina 252

Uso dei dati di riferimento

La sezione descrive i dati di riferimento e le operazioni da eseguire affinché sia possibile utilizzarli.

Dati di riferimento

I dati di riferimento sono costituiti dagli indirizzi utilizzati da DB2SE_USA_GEOCODER per convertire singoli indirizzi in coordinate. Questi dati sono costituiti dagli indirizzi più recenti raccolti dall'ufficio di censimento degli Stati Uniti. Quando il DB2SE_USA_GEOCODER legge un indirizzo nel database, ricerca tra i dati di riferimento:

- I nomi di determinate strade nell'area indicata dal codice postale dell'indirizzo. Il geocoder cerca i nomi che corrispondono al nome della strada dell'indirizzo in base al livello di corrispondenza indicato, o in base a un livello superiore; ad esempio 80% o superiore.
- Gli indirizzi che corrispondono al numero nell'indirizzo.

Se viene riscontrata una corrispondenza che non ha il livello richiesto, il geocoder restituisce le coordinate dell'indirizzo individuato. In caso contrario, il geocoder non restituisce alcun valore.

Per influenzare l'elaborazione eseguita dal geocoder DB2SE_USA_GEOCODER, è possibile utilizzare un file di configurazione avanzato definito *file indicatore di posizione*. Utilizzando la configurazione predefinita di DB2® Spatial Extender non sarà necessario modificare il file.

Impostazione dell'accesso ai dati di riferimento

I dati di riferimento per DB2SE_USA_GEOCODER si trovano in uno dei CD di Spatial Extender. In questa sezione viene descritta la procedura a cui attenersi per accedere ai dati.

Procedura:

Per accedere ai dati di riferimento predefiniti del geocoder, procedere come segue:

1. Stabilire se lasciare i dati di riferimento sul CD o se memorizzarli sul disco fisso. Lasciando i dati sul CD, si eviterà di occupare spazio per circa 700 MB sul disco, mentre copiandoli sul disco, verranno richiamati più velocemente durante le operazioni.
2. Se si desidera memorizzare i dati di riferimento sul disco fisso, procedere come segue:
 - a. Verificare che lo spazio sul disco fisso sia sufficiente per contenere i dati (circa 700 MB).
 - b. Copiare i dati sul disco fisso. Le istruzioni per questa operazione sono riportate nel file README che accompagna i dati di riferimento.
 - c. Verificare che i dati siano stati copiati correttamente. In ambiente UNIX per verificare il corretto caricamento dei dati, controllare nella directory `$DB2INSTANCE/sqlib/gse/refdata/`. Per verificare il corretto caricamento dei dati in Windows NT controllare nella directory `%DB2PATH%\gse\refdata\`.
3. Indicare al DB2SE_USA_GEOCODER il nome e l'ubicazione del file indicatore di posizione e la carta geografica di base, impostando i parametri `base_map` e `locator_file` di DB2SE_USA_GEOCODER sui valori appropriati. Per ulteriori informazioni, rivolgersi al responsabile del database o al rappresentante IBM.

Registrazione di un geocoder

DB2SE_USA_GEOCODER viene registrato in DB2 Spatial Extender automaticamente quando un database viene abilitato per le operazioni spaziali. E' necessario registrare tutti i geocoder che si desidera utilizzare.

Prerequisiti:

Per registrare un geocoder, è necessario che l'ID utente utilizzato abbia l'autorizzazione SYSADM o DBADM per il database in cui si trova il geocoder.

Procedura:

E' possibile registrare un geocoder nei seguenti modi:

- Eseguire la registrazione dalla finestra Registra geocoder del Centro di controllo DB2.
- Lanciare il comando **db2se register_gc**.
- Eseguire un'applicazione che richiama la procedura memorizzata `db2gse.ST_register_geocoder`.

Argomenti correlati:

- "Geocoder e funzioni di geocoding" a pagina 93

Parte 3. Creazione di progetti che utilizzano dati spaziali

Capitolo 8. Impostazione delle risorse spaziali per un progetto

Dopo aver abilitato il database per le operazioni spaziali, è possibile creare progetti che utilizzano dati spaziali. Le risorse necessarie per ciascun progetto sono un sistema di coordinate a cui associare i dati spaziali, e un sistema di riferimento spaziale che definisce l'estensione dell'area geografica a cui i dati fanno riferimento. Il presente capitolo:

- Descrive la natura dei sistemi di coordinate e spiega come crearli
- Descrive la natura dei sistemi di riferimento spaziali e spiega come crearli

Uso dei sistemi di coordinate

Se si desidera sviluppare un progetto che utilizza dati spaziali, è necessario stabilire se i dati devono fare riferimento ad uno dei sistemi di coordinate registrati nel catalogo di Spatial Extender. Se nessuno dei sistemi di coordinate soddisfa le caratteristiche richieste, è possibile creare un nuovo sistema. Questa sezione descrive i sistemi di coordinate ed introduce le attività di selezione e di creazione dei sistemi.

Sistemi di coordinate

Un sistema di coordinate è una struttura che definisce la posizione degli oggetti in una determinata area; ad esempio un'area sulla superficie terrestre o l'intera superficie terrestre. DB2[®] Spatial Extender supporta i seguenti tipi di sistemi di coordinate per determinare la posizione di una caratteristica geografica:

Sistema di coordinate geografiche

Un *sistema di coordinate geografiche* è un sistema di riferimento (consultare la sezione "Sistemi di riferimento spaziali" a pagina 68) che utilizza una superficie sferica tridimensionale per individuare una posizione sulla Terra. Qualsiasi posizione sulla Terra può essere indicata da un punto delle coordinate di longitudine e latitudine basate sulle unità angolari della misura.

Sistema di coordinate proiettate

Il *sistema di coordinate proiettate* è una rappresentazione piana, bidimensionale della Terra. Utilizza le coordinate lineari (Cartesiane) basate sulle unità di misura lineari. Si basa su un modello sferico (o sferoidale) della Terra, e le coordinate utilizzate sono relative alle coordinate geografiche mediante una trasformazione di proiezione.

Argomenti correlati:

- "Sistema di coordinate geografiche" a pagina 59
- "Sistemi di coordinate proiettate" a pagina 65

Riferimenti correlati:

- "Sistemi di coordinate supportati" a pagina 525

Sistema di coordinate geografiche

Un *sistema di coordinate geografiche* è un sistema di riferimento (consultare la sezione "Sistemi di riferimento spaziali" a pagina 68) che utilizza una superficie sferica tridimensionale per individuare una posizione sulla Terra. Qualsiasi posizione sulla

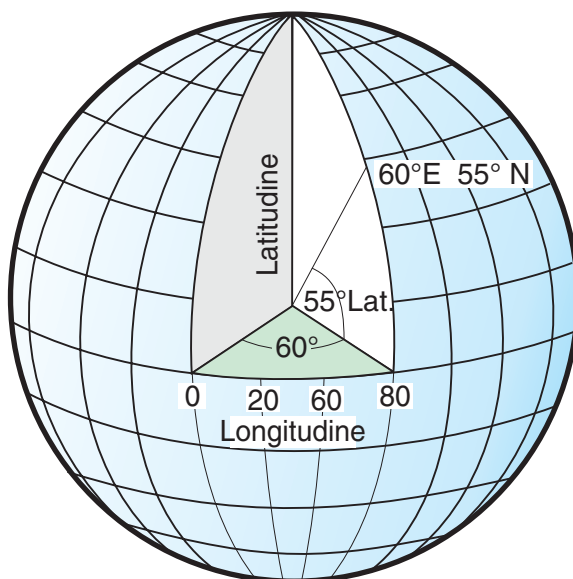
Impostazione delle risorse spaziali per un progetto

Terra può essere indicata da un punto delle coordinate di longitudine e latitudine. I valori dei punti possono essere espressi nelle seguenti unità di misura:

- Unità lineari quando il sistema di coordinate geografiche ha un ID di sistema di riferimento spaziale (SRID) che DB2[®] Geodetic Extender è in grado di riconoscere.
- Una delle seguenti unità quando il sistema di coordinate geografiche ha un SRID che DB2 Geodetic Extender non è in grado di riconoscere:
 - Grado decimale
 - Minuto decimale
 - Secondi decimali
 - Gradianti
 - Radianti

Per gli intervalli di valori consentiti per queste unità, consultare la sezione “Sistemi di coordinate supportati” a pagina 525.

Ad esempio, la Figura 6 mostra un sistema di coordinate geografiche in cui una posizione è dalle coordinate longitudine 80 gradi Est e latitudine 55 gradi Nord.

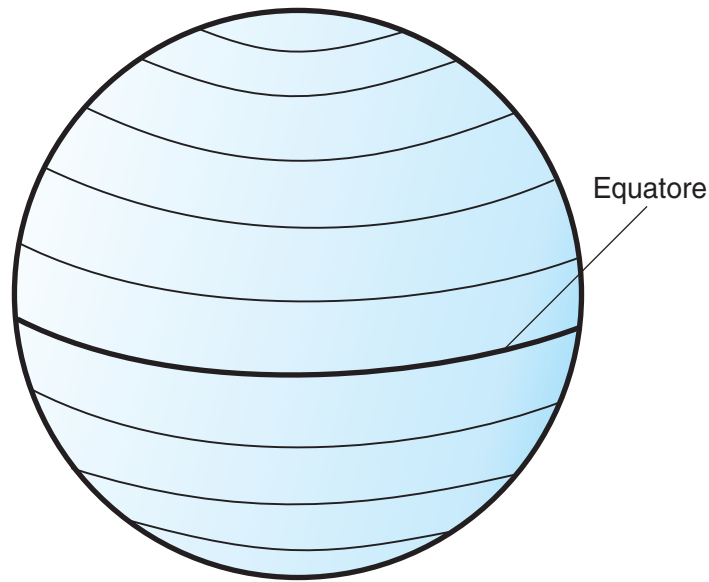


Valori di longitudine e latitudine

Figura 6. Sistema di coordinate geografiche

Le linee che vanno da est a ovest hanno una latitudine costante e sono chiamate *paralleli*. Sono equidistanti e parallele tra loro e formano cerchi concentrici attorno alla Terra. L'*equatore* è la circonferenza più grande e divide la Terra a metà. E' equidistante dai due poli e il valore di questa linea di latitudine è zero. Le posizioni a nord dell'equatore hanno valori di latitudine positivi che vanno da 0 a +90 gradi, mentre le posizioni a sud dell'equatore hanno valori di latitudine negativi che vanno da 0 a -90 gradi.

La Figura 7 mostra le linee di latitudine.

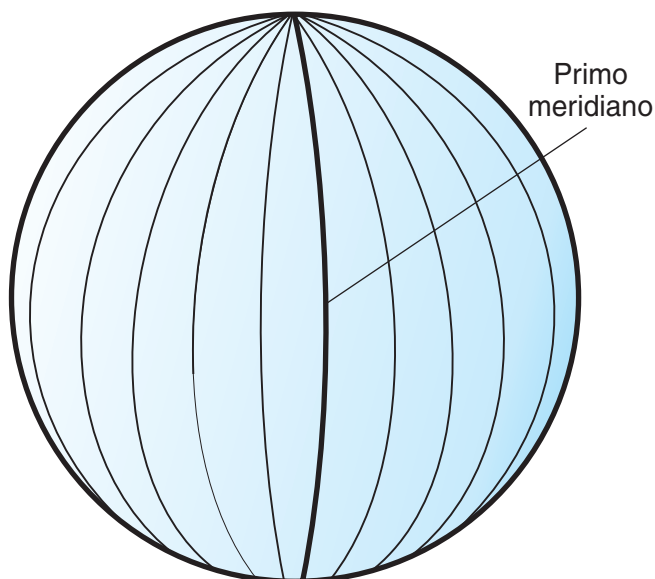


Paralleli (Linee di latitudine)

Figura 7. Linee di latitudine

Le linee che vanno da nord a sud hanno una longitudine costante e sono chiamate *meridiani*. Formano dei cerchi della stessa misura della Terra e si intersecano con i poli. Il *primo meridiano* è la linea di longitudine che definisce l'origine (zero gradi) delle coordinate di longitudine. Il primo meridiano più comune e utilizzato è costituito dalla linea che passa per Greenwich, Inghilterra. Esistono tuttavia, altri meridiani che sono stati utilizzati come primo meridiano, ad esempio quelli che passano per Berna, Bogota, e Parigi. Le posizioni a est del primo meridiano fino al rispettivo meridiano *antipodale* (la continuazione del primo meridiano dall'altro lato del globo), hanno longitudine positiva il cui valore varia tra 0 e +180 gradi. Le posizioni a ovest del primo meridiano hanno longitudine negativa il cui valore varia tra 0 e -180 gradi.

La Figura 8 a pagina 62 mostra le linee di longitudine.



Meridiani (Linee di longitudine)

Figura 8. Linee di longitudine

La latitudine e la longitudine ricoprono il globo formando una griglia definita *reticolato*. Il punto di origine del reticolato è (0,0), ovvero il punto in cui si intersecano il primo meridiano e l'equatore. L'equatore è l'unico punto del reticolato in cui la distanza lineare corrispondente a un grado di latitudine è approssimativamente uguale alla distanza corrispondente a un grado di longitudine. Poiché le linee di longitudine convergono ai poli, la distanza tra ciascun meridiano è diversa all'altezza di ogni parallelo. Quindi avvicinandosi ai poli, la distanza corrispondente a un grado di latitudine sarà molto più grande a quella corrispondente a un grado di longitudine.

E' difficile determinare la lunghezza delle linee di latitudine attraverso il reticolato. Le linee di latitudine sono cerchi concentrici che diventano più piccoli avvicinandosi ai poli. Formano un solo punto ai poli dove hanno inizio i meridiani. All'equatore un grado di longitudine è circa 111,321 chilometri mentre a 60 gradi di latitudine, un grado di longitudine è solo 55,802 chilometri (questo calcolo approssimativo si basa sullo sferoide di Clarke 1866). Quindi, poiché non esiste una lunghezza uniforme tra gradi di latitudine e longitudine, la distanza tra punti non può essere misurata in modo uniforme utilizzando unità di misura angolari.

La Figura 9 a pagina 63 mostra le diverse dimensioni delle posizioni nel reticolato.

Impostazione delle risorse spaziali per un progetto

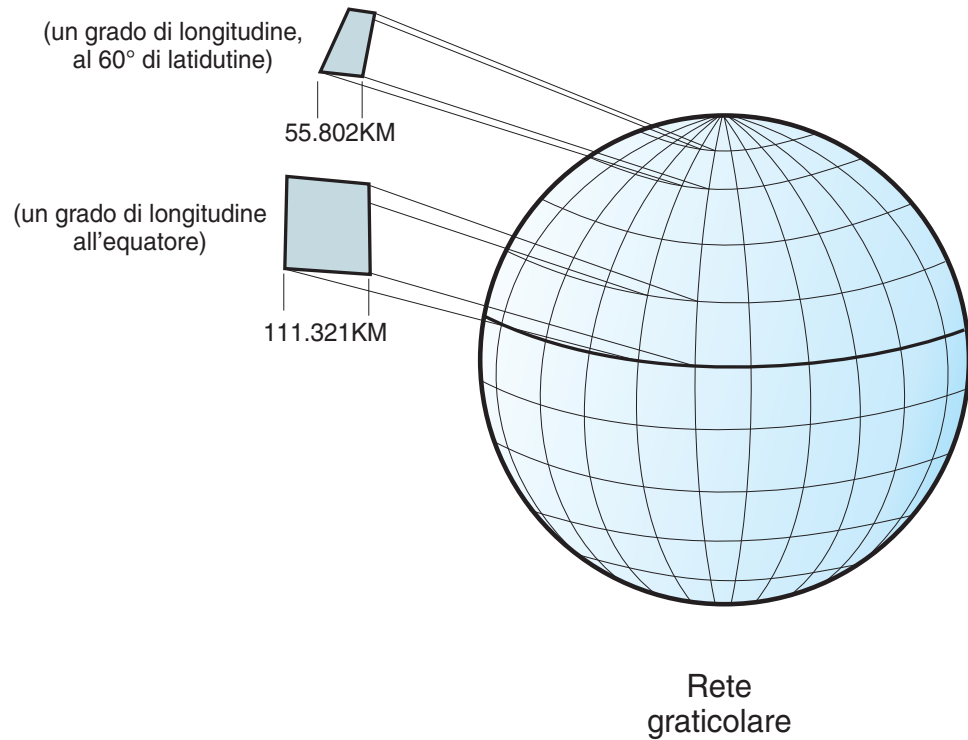


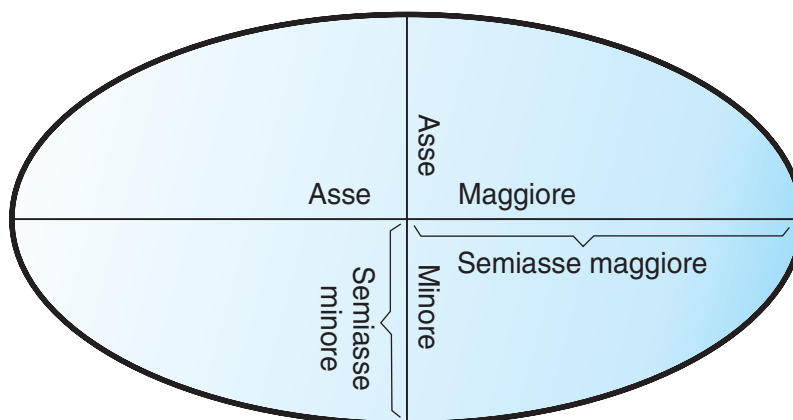
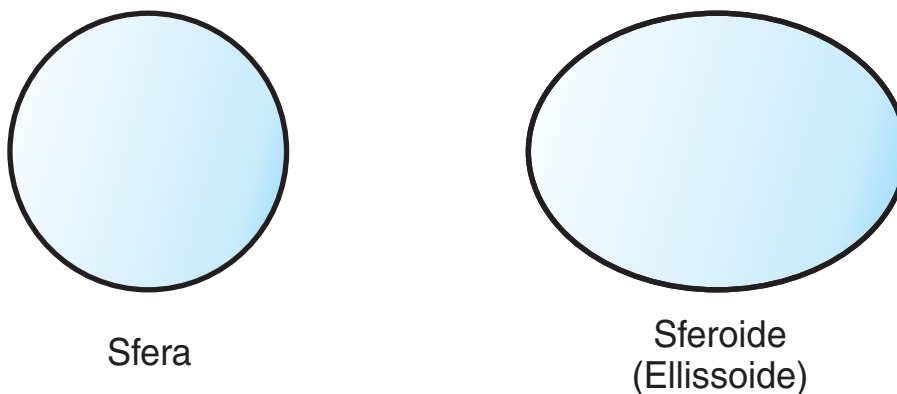
Figura 9. Differenti dimensioni tra le posizioni nel reticolato

Un sistema di coordinate può essere definito da una approssimazione sferica o sferoidale della forma della Terra. Dal momento che la Terra non è perfettamente rotonda, le carte geografiche scala risulteranno più precise se basate su uno sferoide, in base alla posizione sulla Terra. La forma *sferoide* si basa su un'ellisse, così come una sfera si basa su un cerchio.

La forma dell'ellisse è determinata dai due raggi. Il raggio più lungo, è chiamato asse semimaggiore, mentre il più corto asse semiminore. Un ellissoide è una figura geometrica tridimensionale formata dalla rotazione di un'ellisse attorno ad uno dei suoi assi.

La Figura 10 a pagina 64 mostra la sfera e lo sferoide che rappresentano la Terra e gli assi maggiore e minore di un'ellisse.

Impostazione delle risorse spaziali per un progetto



Gli assi maggiore e minore di un'ellisse

Figura 10. Sfera e sferoide

Un *datum* è una serie di valori che definiscono la posizione dello sferoide che rappresenta il centro della Terra. Il datum fornisce una cornice di riferimento per misurare le posizioni e definire l'origine e l'orientamento delle linee di latitudine e di longitudine. Alcuni datum sono globali e cercano di fornire una buona media di precisione per qualsiasi punto sulla Terra. Un datum locale allinea il proprio sferoide alla superficie della Terra in un'area particolare. In tal modo, le misure del sistema di coordinate non saranno precise se utilizzate in un'area diversa da quella alla quale si riferiscono. Per ulteriori informazioni sugli ellissoidi, consultare la sezione "Sistemi di coordinate supportati" a pagina 525.

La Figura 11 a pagina 65 mostra i diversi allineamenti dei datum con la superficie terrestre. In questa particolare posizione, il datum locale, NAD27, è più simile alla superficie della Terra, del datum centrato, WGS84.

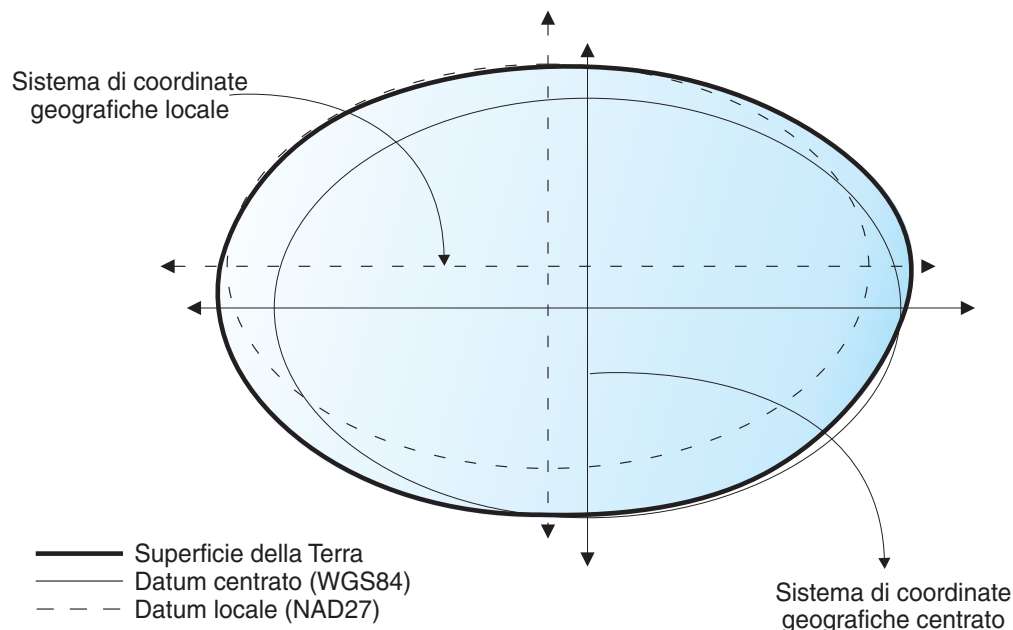


Figura 11. Allineamenti dei datum

Quando si modifica il datum, il sistema di coordinate geografiche ne risente e i valori delle coordinate cambiano di conseguenza. Ad esempio, le coordinate in DMS di un punto di controllo in a Redlands, California utilizzando il datum North American del 1983 (NAD 1983) sono: "-117 12 57.75961 34 01 43.77884". Le coordinate dello stesso punto nel datum North American del 1927 (NAD 1927) sono: "-117 12 54.61539 34 01 43.72995".

Sistemi di coordinate proiettate

Il sistema di coordinate proiettate è una rappresentazione piana, bidimensionale della Terra. Si basa su un sistema di coordinate geografiche sferico o sferoidale, ma utilizza, per le coordinate, unità di misura lineari in modo che il calcolo della distanza e dell'area possa essere facilmente eseguito utilizzando le stesse unità di misura.

Nella proiezione piana, le coordinate di latitudine e longitudine sono convertite in coordinate x, y . La coordinata x rappresenta generalmente la direzione verso est di un punto, mentre la coordinata y ne rappresenta la direzione verso nord. La linea centrale che va da est a ovest viene definita asse x , e la linea che va da nord a sud, asse y .

L'intersezione dell'asse x e dell'asse y rappresenta l'origine e generalmente ha coordinate $(0,0)$. I valori nella parte superiore all'asse x sono positivi, quelli inferiori all'asse x sono negativi. Le linee parallele all'asse x sono equidistanti tra loro. I valori a destra dell'asse y sono positivi, quelli a sinistra sono negativi. Le linee parallele all'asse y sono equidistanti.

La conversione di un sistema di coordinate geografiche tridimensionale in un sistema di coordinate proiettate piano bidimensionale, viene effettuata mediante formule matematiche. Tale trasformazione viene definita *proiezione della mappa*. Le proiezioni della mappa, generalmente vengono classificate in base alla superficie di proiezione utilizzata, ad esempio, conica, cilindrica o piana. Le proprietà dell'oggetto rappresentato possono apparire distorte in base al tipo di proiezione

Impostazione delle risorse spaziali per un progetto

utilizzata. Le proiezioni sono progettate per minimizzare la distorsione di una o due caratteristiche dei dati, tuttavia la distanza, l'area, la forma, la direzione o una combinazione di tutte queste proprietà potrebbero non essere una rappresentazione precisa dei dati che si desidera modellare. Esistono diversi tipi di proiezioni. Alcune carte geografiche tendono a preservare la precisione delle proprietà geografiche, altre invece tendono a minimizzare la distorsione complessiva, come ad esempio la proiezione *Robinson*. I tipi più comuni di proiezione includono:

Proiezioni equal area

In questo tipo di proiezioni, l'area della caratteristica geografica, viene rispettata, mentre la forma, l'angolo e la scala risulteranno distorti. La proiezione conica *Albers Equal Area* è un esempio di proiezione equal area.

Proiezioni conformi

In questo tipo di proiezioni viene rispettata la forma locale delle aree di piccole dimensioni e i singoli angoli che rappresentano le relazioni spaziali, mostrando le linee del reticolato verticali che si intersecano con angoli di 90 gradi sulla carta geografica. Vengono rispettate le dimensioni di tutti gli angoli, tuttavia l'area della carta geografica appare distorta. Le proiezioni *Mercator* e *Lamber Conformal Conic* sono esempi di proiezioni conformi.

Proiezioni equidistanti

In questo tipo di proiezione vengono rispettate le distanze tra certi punti applicando la scala relativa a una determinata serie di dati. Alcune distanze rappresenteranno esattamente le stesse distanze sulla Terra in base a una determinato fattore di scala. Il fattore di scala risulterà impreciso se applicato esternamente alla serie di dati. Le proiezioni *Sinusoidale* e *Equidistant Conic* sono esempi di proiezioni equidistanti.

Proiezioni true-direction o azimuthal

In questo tipo di proiezioni viene rispettata la direzione da un punto all'altro mantenendo alcuni degli archi del circolo maggiore. Tali proiezioni indicano correttamente la direzione o gli azimuth di tutti i punti sulla carta geografica rispetto al centro. Le carte azimutali posso essere combinate alle proiezioni equal area, conformi ed equidistanti. Le proiezioni *Lambert Equal Area Azimuthal* e *Azimuthal Equidistant* sono esempi di proiezioni azimutali.

Argomenti correlati:

- "Sistema di coordinate geografiche" a pagina 59
- "Sistemi di coordinate" a pagina 59

Attività correlate:

- "Scrittura di applicazioni per DB2 Spatial Extender" a pagina 133

Riferimenti correlati:

- "Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti" a pagina 125
- "ST_create_coordsys" a pagina 236
- "Sistemi di coordinate supportati" a pagina 525

Selezione o creazione dei sistemi di coordinate

Una volta abilitato un database per le operazioni spaziali, è possibile avviare progetti basati su dati spaziali. Il primo passo della pianificazione di un progetto è di stabilire il sistema di coordinate da utilizzare. Sono disponibili le seguenti opzioni:

Impostazione delle risorse spaziali per un progetto

- E' possibile utilizzare il sistema di coordinate fornito con DB2 Spatial Extender oppure un sistema creato da un utente. DB2 Spatial Extender viene fornito con più di 2000 sistemi di coordinate. Alcuni di questi sistemi sono:
 - Un sistema di coordinate che DB2 Spatial Extender indica come “Non specificato”. Utilizzare questo sistema quando:
 - Si desidera definire delle posizioni che non hanno un diretto contatto con la superficie terrestre; ad esempio, la posizione di uffici in un edificio o di scaffali in un magazzino.
 - E' possibile definire tali posizioni in termini di coordinate positive che comprendono pochi valori decimali o nessuno.
 - GCS_NORTH_AMERICAN_1983. Utilizzare questo sistema di coordinate se si desidera definire delle posizioni negli Stati Uniti; ad esempio:
 - Durante l'importazione dei dati spaziali riferiti agli Stati Uniti dal CD “Carte geografiche e dati” fornito con DB2 Spatial Extender.
 - Se si desidera utilizzare il geocoder fornito con DB2 Spatial Extender per eseguire il geocoding degli indirizzi degli Stati Uniti

Per ulteriori informazioni su questi sistemi di coordinate, sugli altri sistemi forniti con DB2 Spatial Extender e su eventuali sistemi creati da altri utenti, consultare la vista del catalogo DB2SE.ST_COORDINATE_SYSTEMS.
- E' possibile creare un proprio sistema di coordinate.

Prerequisiti:

Per creare un sistema di coordinate, è necessario che l'ID utente utilizzato disponga di autorizzazione SYSADM o DBADM per il database abilitato alle operazioni spaziali. Per utilizzare un sistema di coordinate esistente non è richiesta alcuna autorizzazione.

Procedura:

E' possibile creare un sistema di coordinate nei seguenti modi:

- Dalla finestra di creazione dei sistemi di coordinate del Centro di controllo DB2.
- Con il comando **db2se create_cs** da un processore riga comandi db2se.
- Eseguendo un'applicazione che richiami la procedura memorizzata db2se.ST_create_coordsys.

Argomenti correlati:

- “Sistemi di coordinate” a pagina 59

Attività correlate:

- “Scrittura di applicazioni per DB2 Spatial Extender” a pagina 133

Riferimenti correlati:

- “Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125
- “ST_create_coordsys” a pagina 236

Impostazione dei sistemi di riferimento spaziali

Se si desidera sviluppare un progetto che utilizza dati spaziali, è necessario stabilire se è possibile utilizzare uno dei sistemi di riferimento spaziali disponibili. Se nessuno dei sistemi disponibili è appropriato, è possibile creare un nuovo sistema. Questa sezione descrive i sistemi di riferimento spaziali e introduce le attività di selezione e di creazione dei sistemi.

Sistemi di riferimento spaziali

Un *sistema di riferimento spaziale* è costituito da una serie di parametri che comprende:

- Il nome del sistema di coordinate dal quale provengono le coordinate.
- L'identificativo numerico che identifica unicamente il sistema di riferimento spaziale.
- Le coordinate che definiscono la massima estensione di uno spazio possibile, in base a un intervallo di coordinate dato.
- I numeri utilizzati nelle operazioni matematiche per convertire le coordinate ricevute come input in valori che è possibile elaborare.

Le seguenti sezioni trattano dei valori dei parametri che definiscono un identificativo, un'estensione massima di spazio e i fattori di conversione.

Identificativo del sistema di riferimento spaziale:

L'identificativo del sistema di riferimento spaziale (SRID) viene utilizzato come parametro input per diverse funzioni spaziali.

Per un sistema di riferimento spaziale geodetico, il valore SRID deve essere compreso tra 2000000000 e 2000001000. DB2® Geodetic Extender fornisce 318 sistemi di riferimento spaziali geodetici predefiniti (SRS). Per ulteriori informazioni, consultare la sezione "DB2 Geodetic Extender" a pagina 157.

Definizione dello spazio che racchiude le coordinate memorizzate in una colonna spaziale:

Le coordinate contenute in una colonna spaziale definiscono delle aree che si estendono in diverse zone della Terra. Lo spazio compreso dall'area—da est ad ovest e da nord a sud—è definito *estensione spaziale*. Come esempio, si consideri una pianura soggetta ad inondazioni le cui coordinate sono memorizzate in una colonna spaziale. Si supponga che le estremità ovest ed est abbiano rispettivamente una latitudine di -24,556 e -19,338, e che la longitudine delle estremità a nord e a sud sia rispettivamente 18,819 e 15,809 gradi. L'estensione spaziale dell'area sarà uno spazio piano compreso tra le due latitudini e le due longitudini. Assegnando a questi valori determinati parametri, sarà possibile includerli in un sistema di riferimento spaziale. Se la colonna spaziale comprende coordinate e misure Z, il sistema di riferimento dovrà contenere anche i valori massimi e minimi delle coordinate e misure Z.

Il termine *estensione spaziale* può riferirsi non solo ad un'area esistente, simile a quella descritta nel paragrafo precedente, ma anche ad un'area potenziale. Se, ad esempio, è prevista una ulteriore estensione della pianura descritta nell'esempio precedente nei successivi cinque anni, sarà possibile calcolare quali saranno le coordinate delle estremità occidentale, orientale, settentrionale e meridionale al termine dei cinque anni. I valori previsti potranno quindi essere assegnati, oltre

alle coordinate attuali, ai parametri dell'estensione spaziale. In tal modo, il sistema di riferimento spaziale verrà periodicamente aggiornato con l'ampliarsi dell'area, e i nuovi valori di latitudine e longitudine vengono aggiunti automaticamente alla colonna spaziale. Se il sistema di riferimento spaziale è limitato alla latitudine e alla longitudine originale, sarà necessario modificarlo o sostituirlo ogni qual volta l'estensione dell'area aumenta.

Conversione in valori per il miglioramento delle prestazioni:

Generalmente, la maggior parte delle coordinate di un sistema, sono espresse in valori decimali, e le altre sono numeri interi. Inoltre, le coordinate che si trovano ad est dell'origine, hanno valore positivo, e quelle a sinistra valore negativo. Prima di essere memorizzate da Spatial Extender, le coordinate negative vengono convertite in valori positivi e i valori decimali in numeri interi. Di conseguenza, tutte le coordinate vengono memorizzate da Spatial Extender come numeri interi. Lo scopo di tale conversione, è di migliorare le prestazioni dell'applicazione durante l'elaborazione delle coordinate.

Nei sistemi di riferimento spaziali, alcuni parametri vengono utilizzati per effettuare le conversioni descritte nel paragrafo precedente. Il parametro definito *valore di scarto*, viene sottratto da ciascuna coordinata negativa lasciando come risultato un valore positivo. Ciascuna coordinata decimale viene moltiplicata per un altro parametro, definito *fattore di scala*, in modo da ottenere come risultato un numero intero con un valore di precisione identico a quello del valore decimale. L'offset viene sottratto sia dalle coordinate positive che negative, e la moltiplicazione per il fattore di scala viene effettuata per le coordinate non decimali, così come per quelle decimali. In tal modo le coordinate positive e non decimali manterranno le stesse proporzioni rispetto a quelle negative e decimali.

Tali conversioni vengono effettuate internamente all'applicazione e sono valide fino a quando le coordinate non vengono nuovamente richiamate. I risultati delle interrogazioni e degli input contengono sempre le coordinate nel formato originale.

Argomenti correlati:

- "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73
- "Sistemi di coordinate" a pagina 59

Attività correlate:

- "Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo" a pagina 69
- "Creazione di un sistema di riferimento spaziale" a pagina 75

Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo

Una volta aver stabilito il sistema di coordinate da utilizzare, scegliere un sistema di riferimento spaziale adatto ai propri dati. DB2 Spatial Extender fornisce cinque sistemi di riferimento spaziali per i dati spaziali, mentre DB2 Geodetic Extender fornisce 318 sistemi di riferimento spaziali geodetici per i dati geodetici.

Procedura:

Impostazione delle risorse spaziali per un progetto

Per stabilire se è possibile utilizzare uno dei sistemi di riferimento spaziali o geodetici predefiniti, procedere come segue:

1. Rispondere alle seguenti domande:
 - Il sistema di coordinate su cui si basa il sistema di riferimento spaziale predefinito, copre l'area con cui si desidera lavorare?
Questi sistemi di coordinate sono riportati nella sezione "Sistemi di riferimento spaziali forniti con DB2 Spatial Extender" a pagina 71.
 - I dati si trovano in un sistema di coordinate geografiche che utilizza gradi decimali o gradi come unità di misura? I dati occupano una vasta parte della superficie della Terra? Si desidera effettuare calcoli precisi di distanze, lunghezze ed aree? Alcuni dei dati utilizzati si trovano in prossimità del Polo nord, Polo Sud o la linea di separazione data internazionale?
Se la risposta a una qualsiasi di queste domande è sì, è preferibile utilizzare uno dei 318 sistemi di riferimento spaziali geodetici predefiniti. Per informazioni su questi sistemi, consultare la sezione "Datum supportati da DB2 Geodetic Extender" a pagina 211.
 - I fattori di conversione associati a uno dei sistemi di riferimento spaziali predefiniti, è adatto ai dati di coordinate utilizzati?
Spatial Extender utilizza valori di *scarto* e fattori di *scala* per convertire i dati di coordinate forniti in numeri interi positivi. Per stabilire se i dati di coordinate funzioneranno con i valori di scarto e i fattori di scala forniti per uno dei sistemi di riferimento spaziali predefiniti, procedere come segue:
 - a. Fare riferimento alle informazioni riportate nella sezione "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73.
 - b. Si osservi come questi fattori sono definiti per i sistemi di riferimento spaziali predefiniti. Se, dopo aver applicato il valore di scarto alle coordinate X e Y minime, tali coordinate non sono entrambi maggiori di 0, sarà necessario creare un nuovo sistema di riferimento spaziale e definire manualmente i valori di scarto. Per ulteriori informazioni sulla creazione di un nuovo sistema di riferimento spaziale, consultare la sezione "Creazione di un sistema di riferimento spaziale" a pagina 75.
 - I dati utilizzati comprendono le coordinate relative ad altezza e profondità (coordinate Z) o a misure (coordinate M)?
Se si utilizzano coordinate Z o M, potrebbe essere necessario creare un nuovo sistema di riferimento spaziale con fattori di scala e di scarto Z o M adatti ai propri dati.
2. Se i sistemi di riferimento spaziali esistenti o i sistemi di riferimento geodetico non sono adatti ai propri dati, sarà necessario "Creazione di un sistema di riferimento spaziale" a pagina 75.

Dopo aver stabilito qual è il sistema di riferimento spaziale più adatto ai propri dati, specificarlo in Spatial Extender durante lo svolgimento di una delle seguenti attività:

- "Creazione di colonne spaziali" a pagina 83
- "Registrazione delle colonne spaziali" a pagina 85

Argomenti correlati:

- "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73
- "Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender" a pagina 158
- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- “Creazione di un sistema di riferimento spaziale” a pagina 75
- “Creazione di colonne spaziali” a pagina 83
- “Registrazione delle colonne spaziali” a pagina 85
- “Creazione di un sistema di riferimento spaziale: Guida di Spatial Extender”
- “Registrazione di una colonna spaziale con un sistema di riferimento spaziale: Guida di Spatial Extender”
- “Selezione di un sistema di riferimento spaziale: Guida di Spatial Extender”

Riferimenti correlati:

- “Sistemi di riferimento spaziali forniti con DB2 Spatial Extender” a pagina 71
- “Datum supportati da DB2 Geodetic Extender” a pagina 211

Sistemi di riferimento spaziali forniti con DB2 Spatial Extender

DB2 Spatial Extender fornisce i sistemi di riferimento spaziali indicati nella seguente tabella, i sistemi di coordinate su cui si basa ciascun sistema e i valori di e i fattori di scala utilizzati da DB2 Spatial Extender per convertire i dati di coordinate in numeri interi positivi. E' possibile reperire informazioni su questi sistemi di riferimento spaziali nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS.

Se si utilizzano gradi decimali (tutti i dati contenuti nel CD dei dati di esempio di DB2 Spatial Extender sono espressi in gradi decimali) i valori di scarto e i fattori di scala per i sistemi di riferimento spaziali predefiniti, supporteranno l'intero intervallo delle coordinate di latitudine-longitudine e manterranno 6 posizioni decimali, che equivalgono approssimativamente a 10 cm.

Se si desidera utilizzare il geocoder che funziona solo con gli indirizzi degli Stati Uniti, selezionare o creare un sistema di riferimento spaziale che gestisca le coordinate degli Stati Uniti,, ad esempio il sistema di coordinate GCS_NORTH_AMERICAN_1983. Se non viene specificato il sistema di coordinate da cui dovranno derivare i dati spaziali, Spatial Extender utilizzerà il sistema di riferimento spaziale DEFAULT_SRS.

Utilizzare la seguente tabella per stabilire se utilizzare un nuovo sistema di riferimento spaziale o crearne uno nuovo (consultare la sezione “Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo” a pagina 69). Se nessuno dei sistemi di riferimento predefiniti è adatto, è possibile creare un nuovo sistema di riferimento spaziale. Per ulteriori informazioni, consultare la sezione “Creazione di un sistema di riferimento spaziale” a pagina 75.

Impostazione delle risorse spaziali per un progetto

Tabella 4. Sistemi di riferimento spaziali forniti con DB2 Spatial Extender

Sistema di riferimento spaziale	ID SRS	Sistema di coordinate	Valori di offset	Fattori di scala	Quando utilizzare
DEFAULT_SRS	0	Nessuno	xOffset = 0 yOffset = 0 zOffset = 0 mOffset = 0	xScale = 1 yScale = 1 zScale = 1 mScale = 1	E' possibile selezionare questo sistema quando i dati non dipendono da un sistema di coordinate o non è necessario specificarne uno.
NAD83_SRS_1	1	GCS_NORTH_AMERICAN_1983	xOffset = -180 yOffset = -90 zOffset = 0 mOffset = 0	xScale = 1,000,000 yScale = 1,000,000 zScale = 1 mScale = 1	E' possibile selezionare questo sistema di riferimento spaziale se si desidera utilizzare i dati di esempio degli Stati Uniti forniti con DB2 Spatial Extender. Se i dati di coordinate utilizzati sono stati raccolti dopo il 1983, utilizzare questo sistema invece di NAD27_SRS_1002.
NAD27_SRS_1002	1002	GCS_NORTH_AMERICAN_1927	xOffset = -180 yOffset = -90 zOffset = 0 mOffset = 0	xScale = 5,965,232 yScale = 5,965,232 zScale = 1 mScale = 1	E' possibile selezionare questo sistema di riferimento spaziale se si desidera utilizzare i dati di esempio degli Stati Uniti forniti con DB2 Spatial Extender. Se i dati di coordinate utilizzati sono stati raccolti prima del 1983, utilizzare questo sistema invece di NAD83_SRS_1.

Tabella 4. Sistemi di riferimento spaziali forniti con DB2 Spatial Extender (Continua)

Sistema di riferimento spaziale	ID SRS	Sistema di coordinate	Valori di offset	Fattori di scala	Quando utilizzare
WGS84_ SRS_1003	1003	GCS_WGS _1984	xOffset = -180 yOffset = -90 zOffset = 0 mOffset = 0	xScale = 5,965,232 yScale = 5,965,232 zScale = 1 mScale = 1	E' possibile selezionare questo sistema di riferimento spaziale se si desidera utilizzare i dati esterni agli Stati Uniti (questo sistema gestisce le coordinate mondiali). Non utilizzare questo sistema se si desidera utilizzare il geocoder predefinito fornito con DB2 Spatial Extender, perché questo geocoder è valido solo per gli indirizzi degli Stati Uniti.
DE_HDN _SRS_1004	1004	GCSW _DEUTSCHE _HAUPTDRE IECKSNETZ	xOffset = -180 yOffset = -90 zOffset = 0 mOffset = 0	xScale = 5,965,232 yScale = 5,965,232 zScale = 1 mScale = 1	Questo sistema di riferimento spaziale si basa su un sistema di coordinate per indirizzi tedeschi.

Argomenti correlati:

- "Sistemi di riferimento spaziali" a pagina 68

Riferimenti correlati:

- "Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS" a pagina 291

Fattori di conversione per la trasformazione delle coordinate in numeri interi

DB2® Spatial Extender utilizza valori di *scarto* e fattori di *scala* per convertire i dati di coordinate forniti in numeri interi positivi. Ai sistemi di riferimento spaziali predefiniti già sono associati valori di scarto e fattori di scala. Se si desidera creare un nuovo sistema di riferimento spaziale, sarà necessario stabilire i fattori di scala e, facoltativamente, i valori di scarto più adatti ai propri dati. Per ulteriori informazioni, consultare la sezione "Creazione di un sistema di riferimento spaziale" a pagina 75.

Valori di scarto

Un valore di scarto è un numero che viene sottratto da tutte le coordinate, lasciando solo numeri positivi come risultato. Spatial Extender converte i dati di coordinate mediante le seguenti formule in modo che tutti i valori modificati risultino maggiori di 0.

Impostazione delle risorse spaziali per un progetto

Nota sulle formule: in queste formule, l'indicazione "min" rappresenta "il minimo tra tutti". Ad esempio, "min(x)" significa "il minimo tra tutte le coordinate x". Il valore di scarto per ciascuna direzione geografica viene rappresentato come *dimensioneOffset*. Ad esempio, *xOffset* è il valore di scarto applicato a tutte le coordinate X.

$$\min(x) - xOffset \geq 0$$

$$\min(y) - yOffset \geq 0$$

$$\min(z) - zoffset \geq 0$$

$$\min(m) - moffset \geq 0$$

Fattori di scala

I fattori di scala sono valori che, se moltiplicati per le coordinate e le misure decimali, danno come risultato numeri interi con almeno lo stesso numero di cifre delle coordinate e misure originali. Spatial Extender converte i dati di coordinate decimali mediante le seguenti formule in modo che tutti i valori modificati risultino numeri interi positivi. I valori convertiti non possono superare 2^{53} (approssimativamente, $9 * 10^{15}$).

Nota sulle formule: in queste formule, l'indicazione "max" rappresenta "il massimo tra tutti". Il valore di scarto per ciascuna dimensione geografica è rappresentato come *dimensioneOffset* (ad esempio, *xOffset* è il valore di scarto applicato a tutte le coordinate X). Il fattore di scala per ciascuna dimensione geografica è rappresentato come *dimensioneScale* (ad esempio, *xScale* è il fattore di scala applicato a tutte le coordinate X).

$$(\max(x) - xOffset) * xScale \leq 2^{53}$$

$$(\max(y) - yOffset) * yScale \leq 2^{53}$$

$$(\max(z) - zoffset) * zScale \leq 2^{53}$$

$$(\max(m) - moffset) * mScale \leq 2^{53}$$

Durante la scelta dei fattori di scala più adatti ai propri dati di coordinate, assicurarsi che:

- Venga utilizzato lo stesso fattore di scala per le coordinate X e Y.
- Se moltiplicato per una coordinata decimale X o Y, il fattore di scala ottenga un valore inferiore a 2^{53} . Una tecnica comune consiste nell'utilizzare un valore di scala che sia una potenza di 10. Ovvero, il fattore di scala dovrebbe essere 10 elevato a uno (10), 10 elevato a due (100), 10 elevato a tre (1000), o, se necessario, un fattore più alto.
- Il fattore di scala deve essere abbastanza grande per assicurare che il numero di cifre del nuovo numero intero sia uguale a quello della coordinata decimale originale.

Esempio:

Si supponga che la funzione ST_Point restituisca un input composto da una coordinata X di 10,01, una coordinata Y di 20,03 e dall'identificativo di un sistema di riferimento spaziale. Richiamando la funzione ST_Point, il valore 10,01 e il valore 20,03 vengono moltiplicati per il fattore di scala del sistema di riferimento spaziale per le coordinate X e Y. Se il fattore di scala è 10, gli interi risultanti memorizzati da Spatial Extender saranno, rispettivamente, 100 e 200. Poiché il numero di cifre in questi valori interi (3) è inferiore al numero di cifre nelle coordinate (4), Spatial Extender non sarà in grado di convertire nuovamente tali interi nel valore delle coordinate originali oppure non sarà in grado di ricavare da essi i valori congruenti con il sistema di coordinate a cui tali coordinate appartengono. Ma se il fattore di scala è 100, i valori interi risultanti memorizzati

da DB2 Spatial Extender saranno 1001 e 2003, valori che possono essere convertiti nuovamente nelle coordinate originali o in valori da cui è possibile derivare coordinate compatibili.

Unità per i valori di scarto e i fattori di scala

Ogni qual volta si utilizza un sistema di riferimento spaziale esistente o se ne crea uno nuovo, le unità per i valori di scarto e i fattori di scala possono variare in base al tipo di sistema di coordinate utilizzato. Ad esempio, se viene utilizzato un sistema di coordinate geografiche, i valori vengono espressi in unità angolari, quali i gradi decimali; se si utilizza un sistema di coordinate proiettate, i valori vengono espressi in unità lineari, ad esempio metri o piedi.

Attività correlate:

- “Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo” a pagina 69

Creazione di un sistema di riferimento spaziale

Se nessuno dei sistemi di riferimento spaziali forniti con DB2 Spatial Extender è adatto ai dati, sarà necessario creare un nuovo sistema di riferimento spaziale.

Procedura:

Per creare un nuovo sistema di riferimento spaziale, procedere come segue:

1. Scegliere l'interfaccia.
E' possibile creare un sistema di riferimento spaziale nei seguenti modi:
 - Utilizzare la finestra Creazione sistema di riferimento spaziale nel Centro di controllo DB2. Consultare la guida in linea per ulteriori informazioni sull'uso di questa finestra.
 - Eseguire il comando **db2se create_srs** dal processore di riga comandi db2se. Per ulteriori informazioni, consultare la sezione “Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125.
 - Eseguendo un'applicazione che richiami la procedura memorizzata db2se.ST_create_srs. Per ulteriori informazioni, consultare la sezione “ST_create_srs” a pagina 238.
2. Specificare uno SRID (spatial reference system ID):
 - Per i dati geodetici in una rappresentazione sferica, specificare un valore SRID compreso tra 200000318 e 2000001000.
 - Per i dati spaziali in una rappresentazione piana, specificare un valore SRID non ancora definito.
3. Stabilire il grado di precisione desiderato. E' possibile::
 - Specificare l'estensione dell'area geografica che verrà utilizzata e i fattori di scala che si desidera utilizzare con i dati di coordinate. Spatial Extender utilizzerà l'estensione specificata e calolerà il valore di scarto.
E' possibile specificare l'estensione in uno dei seguenti modi
 - Scegliere **Estensioni** nella finestra Creazione sistema di riferimento spaziale del Centro controllo.
 - Indicare i parametri appropriati per il comando **db2se create_srs** o la procedura memorizzata db2se.ST_create_srs.
 - Specificare i valori di scarto (richiesti in Spatial Extender per convertire i valori negativi in valori positivi) e i fattori di scala (richiesti in Spatial

Impostazione delle risorse spaziali per un progetto

Extender per convertire i valori decimali in numeri interi). Utilizzare questo metodo se si desiderano risultati molto precisi.

E' possibile specificare i valori di scarto e i fattori di scala in uno dei seguenti modi:

- Scegliere **Offset** nella finestra Creazione sistema di riferimento spaziale del Centro controllo.
- Indicare i parametri appropriati per il comando **db2se create_srs** o la procedura memorizzata **db2se.ST_create_srs**.

Per ulteriori informazioni, consultare la sezione "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73.

4. Calcolare le informazioni di conversione richieste da Spatial Extender per convertire i dati di coordinate in numeri interi positivi, e fornire tali informazioni mediante l'interfaccia scelta. Queste informazioni differiscono in base al metodo scelto al punto 3.
 - Se è stato scelto il metodo "Estensioni" al punto 3, sarà necessario calcolare le seguenti informazioni:
 - Fattori di scala. Se una delle coordinate utilizzate è formata da valori decimali, calcolare i fattori di scala (consultare la sezione "Calcolo dei fattori di scala" a pagina 77). I fattori di scala sono numeri che, se moltiplicati per le coordinate e le misure decimali, danno come risultato numeri interi con almeno lo stesso numero di cifre delle coordinate e misure originali. Se le coordinate sono numeri interi, i fattori di scala possono essere impostati su 1. Se le coordinate sono valori decimali, il fattore di scala dovrebbe essere impostato su un numero che converta la parte decimale in un valore intero. Ad esempio, se le coordinate sono rappresentate in metri e la precisione dei dati è di 1 cm, è necessario un fattore di scala di 100.
 - Valori minimi e massimi per le coordinate e le misure. Per ulteriori informazioni, consultare la sezione "Determinazione delle coordinate e misure minime e massime" a pagina 78.
 - Se è stato scelto il metodo "Offset" al punto 3, sarà necessario calcolare le seguenti informazioni:
 - Valori di scarto
Se i dati di coordinate includono numeri o misure negativi, sarà necessario specificare i valori di scarto che si desidera utilizzare. Un valore di scarto è un numero che viene sottratto da tutte le coordinate, lasciando così solo numeri positivi. Se si utilizzano coordinate positive, impostare tutti i valori di scarto su 0. In caso contrario, selezionare uno scarto che quando applicato ai dati coordinate, ottenga come risultato numeri interi inferiori al valore intero positivo più elevato (9.007.199.254.740.992). Per ulteriori informazioni consultare la sezione "Calcolo dei valori di offset" a pagina 79.
 - Fattori di scala
Se tra le coordinate delle posizioni che si desidera rappresentare vi sono numeri decimali, stabilire i fattori di scala da utilizzare e specificare tali fattori nella finestra Creazione sistema di riferimento spaziale. Consultare la sezione "Calcolo dei fattori di scala" a pagina 77.
5. Eseguire il comando **db2se create_srs** o la procedura memorizzata **db2se.ST_create_srs**.
Ad esempio, il seguente comando crea un sistema di riferimento spaziale chiamato mysrs:

Impostazione delle risorse spaziali per un progetto

```
db2se create_srs miodb -srsName \"miosist\"  
-srsID 100 -xScale 10 -coordsysName  
\"GCS_North_American_1983\"
```

Per ulteriori informazioni sull'esecuzione di un'applicazione che richiami la procedura memorizzata db2se.ST_create_srs, consultare la sezione "ST_create_srs" a pagina 238.

Dopo aver creato il sistema di riferimento spaziale, associarlo a una colonna spaziale mediante una delle seguenti attività:

- "Creazione di colonne spaziali" a pagina 83
- "Registrazione delle colonne spaziali" a pagina 85

Argomenti correlati:

- "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73
- "Sistemi di coordinate" a pagina 59
- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- "Calcolo dei fattori di scala" a pagina 77
- "Determinazione delle coordinate e misure minime e massime" a pagina 78
- "Calcolo dei valori di offset" a pagina 79
- "Creazione di colonne spaziali" a pagina 83
- "Registrazione delle colonne spaziali" a pagina 85
- "Creazione di un sistema di riferimento spaziale: Guida di Spatial Extender"
- "Registrazione di una colonna spaziale con un sistema di riferimento spaziale: Guida di Spatial Extender"

Riferimenti correlati:

- "Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti" a pagina 125
- "ST_create_srs" a pagina 238

Calcolo dei fattori di scala

Se viene creato un sistema di riferimento spaziale e i valori di tutte le coordinate utilizzate sono decimali, calcolare i fattori di scala appropriati per le proprie coordinate e misure. I fattori di scala sono numeri che, se moltiplicati per le coordinate e le misure decimali, danno come risultato numeri interi con almeno lo stesso numero di cifre delle coordinate e misure originali.

Prerequisiti:

Prima di calcolare i fattori di scala che potranno essere utilizzati con i dati, leggere le istruzioni relative alla scelta dei "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73.

Procedura:

Per calcolare i fattori di scala:

Impostazione delle risorse spaziali per un progetto

1. Stabilire quali coordinate X e Y corrispondono esattamente o approssimativamente a un numero decimale. Si supponga ad esempio che tra le varie coordinate X e Y utilizzate, tre sono quelle corrispondenti a numeri decimali: 1,23, 5,1235 e 6,789.
2. Individuare la coordinata decimale con precisione decimale più lunga. Stabilire quindi per quale potenza fattore di 10 è possibile moltiplicare questa coordinata, per ottenere come risultato un numero intero di uguale precisione. Ad esempio, tra le tre coordinate decimali nell'esempio, 5,1235 è quella con precisione decimale più lunga. Moltiplicando questo numero per 10 elevato alla quarta (10000), il risultato sarà 51235, che corrisponde a un numero intero.
3. Verificare se il numero intero prodotto dalla moltiplicazione appena eseguita, è inferiore a 2^{53} . 51235 non è troppo grande. Si supponga comunque che oltre ai valori 1,23, 5,1235 e 6,789, l'intervallo di coordinate X e Y includa un quarto valore decimale, 1000000006,789876. Poiché la precisione decimale di questa coordinata è più lunga della precisione dei primi tre numeri, sarà necessario moltiplicare *questa* coordinata—non 5,1235—per una potenza di 10. Per convertirlo in un numero intero, è possibile moltiplicarlo per 10 elevato a sei (1000000). Ma il valore risultante, 1000000006789876, è maggiore di 2^{53} . Se DB2 Spatial Extender tenterà di memorizzarlo, potrebbero verificarsi risultati imprevisti.

Per evitare questo problema, selezionare una potenza di 10 che, se moltiplicato per la coordinata originale, restituisce un numero decimale che DB2 Spatial Extender potrà ridurre a un numero intero memorizzabile, con perdita minima di precisione. In tal caso, è possibile selezionare 10 elevato a cinque (100000). Moltiplicando 100000 per 1000000006,789876 si ottiene 100000000678987,6. DB2 Spatial Extender arrotonderà questo numero a 100000000678988, riducendone leggermente la precisione.

Dopo aver calcolato i fattori di scala, sarà necessario determinare i valori di estensione (consultare la sezione "Determinazione delle coordinate e misure minime e massime". Quindi eseguire il comando **db2se create_srs** o la procedura memorizzata `db2se.ST_create_srs`.

Argomenti correlati:

- "Fattori di conversione per la trasformazione delle coordinate in numeri interi" a pagina 73
- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- "Determinazione delle coordinate e misure minime e massime" a pagina 78

Determinazione delle coordinate e misure minime e massime

Se si desidera specificare trasformazioni di estensione durante la creazione di un sistema di riferimento spaziale, determinare le coordinate e le misure minime e massime.

Prerequisiti:

Attenersi a questa procedura per la determinazione delle coordinate e delle misure minime e massime se:

- Si desidera creare un nuovo sistema di riferimento spaziale perché nessuno di quelli forniti con DB2 Spatial Extender è adatto ai dati utilizzati. Per ulteriori informazioni, consultare la sezione "Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo" a pagina 69.

- Si desidera utilizzare trasformazioni di estensione per convertire le coordinate.

Procedura:

Per determinare le coordinate e le misure minime e massime delle posizioni che si desidera rappresentare, procedere come segue:

- Determinare le coordinate X minime e massime.

Per rilevare la coordinata X minima, identificare la coordinata X nell'area più a ovest. Se la posizione si trova a ovest del punto di origine, questa coordinata avrà valore negativo. Per rilevare la coordinata X massima, identificare la coordinata X nell'area più a est. Ad esempio, se si desidera rappresentare dei pozzi petroliferi e ciascun pozzo è definito da una coppia di coordinate X e Y, la coordinata che indica la posizione del pozzo più spostata verso ovest, sarà la coordinata minima, mentre la coordinata X che indica la posizione del pozzo più spostata verso est è la coordinata X massima.

- Determinare le coordinate Y minime e massime.

Per rilevare la coordinata Y minima, identificare la coordinata Y nell'area più a sud. Se la posizione dell'oggetto è a sud del punto di origine, la coordinata avrà valore negativo. Per rilevare la coordinata Y massima, identificare la coordinata Y nell'area più a nord.

- Determinare le coordinate Z minime e massime.

La coordinata Z minima è la maggiore delle coordinate di profondità, mentre la coordinata Z massima è la maggiore delle coordinate di altezza.

- Determinare le misure minime e massime

Se si desidera includere unità di misura nei dati spaziali, determinare quale misura ha il valore numerico maggiore e quale l'inferiore.

Per i tipi a multifunzione, ad esempio i multipoligoni, assicurarsi di aver scelto il punto più estremo, del poligono più estremo nella direzione che si sta calcolando. Ad esempio, se si desidera identificare la coordinata X minima, identificare la coordinata X più a ovest del poligono a sua volta più a ovest nel multipoligono.

Dopo aver determinato i valori di estensioni, se una qualsiasi delle coordinate ha valore decimale, sarà necessario calcolare i fattori di scala (consultare la sezione "Calcolo dei fattori di scala" a pagina 77). Altrimenti, eseguire il comando **db2se create_srs** o la procedura memorizzata **db2se.ST_create_srs**.

Attività correlate:

- "Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo" a pagina 69
- "Calcolo dei fattori di scala" a pagina 77

Calcolo dei valori di offset

Se nel sistema di riferimento spaziale creato i dati di coordinate includono numeri o misure negativi, sarà necessario specificare i valori di scarto che si desidera utilizzare. Un valore di scarto è un numero che viene sottratto da tutte le coordinate, lasciando così solo numeri positivi. Quando i valori delle coordinate sono numeri interi positivi invece di numero o misure negative, le prestazioni delle operazioni spaziali migliorano decisamente..

Prerequisiti:

Impostazione delle risorse spaziali per un progetto

Specificare i valori di scarto se i dati di coordinate includono numero o misure negativi.

Procedura:

Per calcolare i valori di scarto per le coordinate utilizzate, procedere come segue:

1. Determinare le coordinate X, Y, e Z negative inferiori nell'intervallo di coordinate delle posizioni che si desidera rappresentare. Se si desidera includere delle misurazioni negative nei dati, determinare i valori inferiori delle misure. Consultare la sezione "Determinazione delle coordinate e misure minime e massime" a pagina 78.
2. Facoltativo ma consigliato: Indicare a DB2 Spatial Extender una estensione superiore a quella che in realtà racchiude tutte le posizioni desiderate. Di conseguenza, dopo aver inserito i dati relativi alle posizioni nella colonna spaziale, sarà possibile aggiungere dati relativi alle posizioni di nuove caratteristiche aggiunte esternamente all'area prevista, senza che sia necessario sostituire il sistema di riferimento con uno più aggiornato.

Per ciascuna coordinata e misura identificate al punto 1, aggiungere una quantità pari al 5 fino al 10 per cento del valore della coordinata o della misura. Il risultato viene definito *valore aumentato*. Ad esempio, se il valore negativo della coordinata X inferiore è -100, è possibile aggiungere -5 ottenendo un valore aumentato di -105. Successivamente, durante la creazione del sistema spaziale di riferimento, bisognerà indicare che la coordinata X inferiore è -105, non il valore originale -100. DB2 Spatial Extender considererà il valore -105 come il limite dell'estremità occidentale dell'estensione.

3. Determinare il valore che sottratto al valore X aumentato, ottenga il risultato zero; questo sarà il valore di scarto delle coordinate X. DB2 Spatial Extender sottrarrà questo numero da tutte le coordinate X in modo da produrre solo valori positivi.

Se ad esempio il valore X incrementato è -105, è necessario sottrarre -105 per ottenere 0. DB2 Spatial Extender sottrarrà quindi -105 da tutte le coordinate X associate alle funzioni rappresentate. Poiché nessuna di queste coordinate è maggiore di -100, tutti i valori risultanti dalla sottrazione saranno positivi.

4. Ripetere il punto 3 per il valore Y aumentato, il valore Z aumentato e la misura aumentata.

Dopo aver calcolato i valori di offset, creare un sistema di riferimento spaziale (consultare la sezione "Creazione di un sistema di riferimento spaziale" a pagina 75).

Attività correlate:

- "Determinazione delle coordinate e misure minime e massime" a pagina 78
- "Creazione di un sistema di riferimento spaziale" a pagina 75

Capitolo 9. Impostazione delle colonne spaziali

Per utilizzare i dati spaziali in un progetto, dopo aver scelto o creato un sistema di coordinate e un sistema di riferimento spaziale, è necessario indicare una o più colonne di una tabella in cui inserire i dati. Il presente capitolo:

- Mette in evidenza i risultati delle interrogazioni delle colonne che possono essere visualizzati graficamente e fornisce istruzioni per la scelta dei tipi di dati per le colonne
- Descrive le attività da svolgere per fornire le colonne
- Descrive le attività da svolgere affinché gli strumenti per la visualizzazione dei dati in forma grafica possano accedere alle colonne

Colonne spaziali

Colonne spaziali con contenuto visualizzabile

Se per interrogare una colonna spaziale viene utilizzato uno strumento di visualizzazione, ad esempio ArcExplorer for DB2[®], i risultati restituiti saranno in formato grafico; ad esempio la mappa di un appezzamento di terreno, o il tracciato di una rete stradale. Alcuni strumenti di visualizzazione richiedono che tutte le righe di una colonna utilizzino lo stesso sistema di riferimento spaziale. Per rispettare tale requisito, è necessario registrare la colonna con un sistema di riferimento spaziale.

Tipi di dati spaziali

Quando si abilita un database per le operazioni spaziali, DB2 Spatial Extender fornisce al database una gerarchia di tipi di dati strutturati. Nella Figura 12 a pagina 82 viene illustrata questa gerarchia. In questa figura, i tipi istanziabili presentano uno sfondo bianco; i tipi non istanziabili presentano uno sfondo ombreggiato.

I tipi di dati istanziabili sono ST_Point, ST_LineString, ST_Polygon, ST_GeomCollection, ST_MultiPoint, ST_MultiPolygon e ST_MultiLineString.

I tipi di dati non istanziabili sono ST_Geometry, ST_Curve, ST_Surface, ST_MultiSurface e ST_MultiCurve.

Impostazione delle colonne spaziali

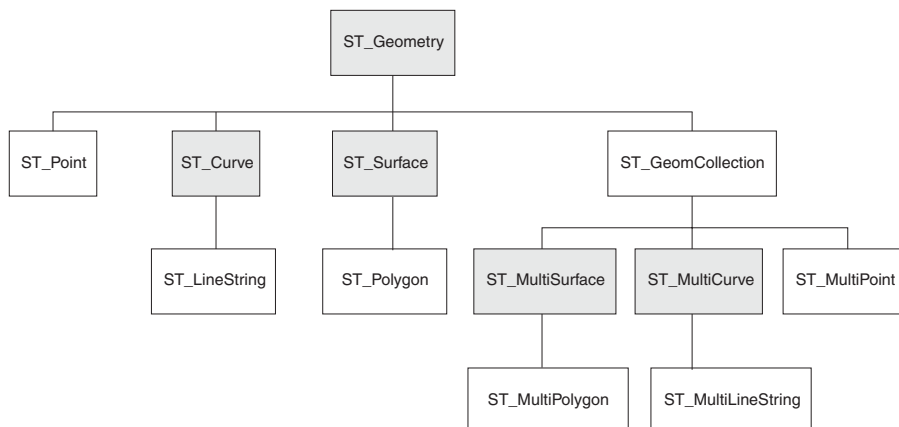


Figura 12. Gerarchia dei tipi di dati spaziali. I tipi di dati compresi nelle caselle bianche sono istanziabili. I tipi di dati compresi nelle caselle ombreggiate non sono istanziabili.

La gerarchia rappresentata nella Figura 12 include:

- Tipi di dati per le caratteristiche geografiche che è possibile assumere come singola unità; ad esempio residenze individuali e laghi isolati.
- Tipi di dati per le funzioni geografiche composti da più unità o componenti; ad esempio, un sistema di canali o un gruppo di isole in un lago.
- Un tipo di dati per le caratteristiche geografiche di tutti i tipi.

Tipi di dati per funzioni a singola unità

Per memorizzare le coordinate che definiscono lo spazio occupato dalle funzioni che è possibile considerare come singole unità, utilizzare ST_Point, ST_LineString e ST_Polygon:

- Utilizzare ST_Point per indicare il punto dello spazio occupato da una caratteristica geografica distinta. La caratteristica può essere di dimensioni molto ridotte, come un pozzo idrico, oppure di dimensioni elevate, come una città; oppure di dimensione intermedia, come un complesso di edifici o un parco. In ogni caso, il punto dello spazio può essere posizionato nel punto di intersezione di una coordinata est-ovest (ad esempio un parallelo) e una coordinata nord-sud (ad esempio un meridiano). Una voce di dati ST_Point comprende una coordinata X e una Y che definiscono tale intersezione. La coordinata X indica il punto di intersezione sulla linea est-ovest; la coordinata Y indica il punto di intersezione sulla linea nord-sud.
- Utilizzare ST_LineString per le coordinate che definiscono lo spazio occupato dalle funzioni lineari; ad esempio le strade e i canali.
- Utilizzare ST_Polygon se si desidera indicare la quantità di spazio coperta da una funzione su più siti; ad esempio un distretto pubblico, una foresta o una zona popolata da animali selvaggi. Una voce di dati ST_Polygon comprende le coordinate che definiscono il contorno di questa funzione.

In alcuni casi è possibile utilizzare ST_Polygon e ST_Point per le stesse funzioni. Si supponga ad esempio che si desideri richiamare informazioni spaziali di un complesso residenziale. Per rappresentare il punto dello spazio in cui è ubicato ciascun edificio del complesso, sarà necessario utilizzare ST_Point per memorizzare le coordinate X e Y che definiscono tale punto. Oppure, se si desidera rappresentare l'area coperta dall'intero complesso, è consigliabile utilizzare ST_Polygon per memorizzare le coordinate che definiscono il contorno dell'area.

Tipi di dati per funzioni a più unità

Utilizzare `ST_MultiPoint`, `ST_MultiLineString` e `ST_MultiPolygon` per memorizzare le coordinate che definiscono gli spazi occupati dalle funzioni composte da più unità:

- Utilizzare `ST_MultiPoint` se si desidera rappresentare caratteristiche geografiche composte da unità di cui tutte posizioni sono indicate da una coordinata X e una Y. Si consideri, ad esempio una tabella contenente righe che rappresentano arcipelaghi. La coordinata X e la coordinata Y di ciascuna isola sono state identificate. Se si desidera includere nella tabella queste coordinate e le coordinate relative all'intero arcipelago, definire una colonna `ST_MultiPoint` in cui inserire tali valori.
- Utilizzare `ST_MultiLineString` se si desidera rappresentare caratteristiche create da unità lineari e memorizzare le coordinate delle posizioni di queste unità e le posizioni di ciascuna caratteristica come un unico valore. Si consideri, ad esempio una tabella contenente righe che rappresentano dei sistemi di corsi d'acqua. Se si desidera includere nella tabella queste coordinate e le coordinate relative ai sistemi e a tutti i rispettivi componenti, definire una colonna `ST_MultiLineString` in cui inserire tali valori.
- Utilizzare `ST_MultiPolygon` se si desidera rappresentare caratteristiche create da unità a più facce e memorizzare le coordinate delle posizioni di queste unità e le posizioni di ciascuna caratteristica come un unico valore. Ad esempio, si consideri una tabella contenente righe che rappresentano zone rurali e le fattorie di ciascuna zona. Se si desidera includere nella tabella le coordinate relative alle zone e alle fattorie, definire una colonna `ST_MultiPolygon` in cui inserire tali valori.

Le unità multiple non sono una raccolta di singole entità, ma si riferiscono ad un aggregato delle parti che costituiscono il tutto.

Un tipo di dati per tutte le funzioni

Se non si è certi sugli altri dati, utilizzare `ST_Geometry`. Poiché `ST_Geometry` è la radice della gerarchia a cui appartengono tutti i tipi di dati, una colonna `ST_Geometry` può contenere lo stesso tipo di voci di dati delle colonne a cui sono stati assegnati altri tipi di dati.

Attenzione:

Se si desidera creare dati per una colonna spaziale utilizzando il geocoder fornito, `DB2SE_USA_GEOCODER`, la colonna dovrà essere di tipo `ST_Point` o `ST_Geometry`. Alcuni strumenti di visualizzazione, tuttavia, non supportano le colonne `ST_Geometry`, ma solo le colonne a cui è stato assegnato un tipo di `ST_Geometry` secondario.

Attività correlate:

- "Registrazione delle colonne spaziali" a pagina 85
- "Creazione di colonne spaziali" a pagina 83

Creazione di colonne spaziali

Questa attività è parte dell'attività "Impostazione delle risorse spaziali per un progetto". Dopo aver scelto un sistema di coordinate e aver determinato il sistema di riferimento spaziale da utilizzare per i dati spaziali, creare una colonna spaziale in una tabella esistente o importare i dati spaziali in una nuova tabella.

Prerequisiti:

Impostazione delle colonne spaziali

Per creare una colonna spaziale, l'ID utente utilizzato deve disporre delle autorizzazioni necessarie per l'istruzione DB2 SQL CREATE TABLE e ALTER TABLE. L'ID utente, inoltre deve disporre di almeno una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database a cui appartiene la tabella che contiene la colonna
- Autorizzazione CREATETAB per il database e privilegio USE per il table space e una delle seguenti:
 - Autorizzazione IMPLICIT_SCHEMA per il database, se lo schema implicito o esplicito dell'indice non esiste
 - Privilegio CREATEIN per lo schema, se il nome dello schema dell'indice si riferisce a uno schema esistente
- Privilegio ALTER per la tabella da modificare
- Privilegio CONTROL per la tabella da modificare
- Privilegio ALTERIN per lo schema della tabella

Procedura:

Per inserire le colonne spaziali nel database, è possibile procedere in diversi modi:

- Creando una tabella che comprenda una colonna spaziale mediante l'istruzione CREATE TABLE di DB2.
- Per aggiungere una colonna spaziale ad una tabella esistente, utilizzare l'istruzione ALTER TABLE di DB2.
- Utilizzando la finestra di creazione indice spaziale del Centro di controllo DB2. Aprire la finestra Colonne spaziali da una tabella. Consultare la guida in linea per ulteriori informazioni sull'uso di questa finestra.
- Se si desidera importare i dati spaziali da un file shape, è possibile creare una tabella che contenga una colonna spaziale in cui memorizzare i dati utilizzando DB2 Spatial Extender. Consultare la sezione "Importazione dei dati shape in una tabella nuova o esistente" a pagina 88.
- Se si desidera importare i dati spaziali da un file di trasferimento SDE, è possibile creare una tabella che contenga una colonna spaziale in cui memorizzare i dati e far sì che gli strumenti di visualizzazione possano accedere alla tabella, utilizzando DB2 Spatial Extender. Consultare la sezione "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90.

Attività successiva: "Registrazione delle colonne spaziali" a pagina 85

Attività correlate:

- "Importazione dei dati shape in una tabella nuova o esistente" a pagina 88
- "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90
- "Registrazione delle colonne spaziali" a pagina 85
- "Scrittura di applicazioni per DB2 Spatial Extender" a pagina 133
- "Creazione di una colonna spaziale: Guida di Spatial Extender"

Riferimenti correlati:

- "ALTER TABLE statement" nella *SQL Reference, Volume 2*
- "CREATE TABLE statement" nella *SQL Reference, Volume 2*

- “Richiamo dei comandi per l’impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125

Registrazione delle colonne spaziali

Se si presenta una delle seguenti situazioni, è preferibile registrare la colonna spaziale:

- Accesso mediante gli strumenti di visualizzazione

Se si desidera generare rappresentazioni grafiche dei dati di una colonna spaziale, utilizzando strumenti di visualizzazione, ad esempio ArcExplorer for DB2, è necessario verificare l’integrità di tali dati impostando un limite che obbliga tutte le righe della colonna a utilizzare lo stesso sistema di riferimento spaziale. Per effettuare tale impostazione, registrare la colonna indicando sia il nome che il sistema di riferimento spaziale utilizzati.

- Accesso mediante gli indici spaziali

Utilizzare lo stesso sistema di coordinate per tutti i dati di una colonna spaziale in cui creare un indice in modo da assicurarsi che l’indice spaziale restituisca i risultati corretti. E’ possibile registrare una colonna spaziale per limitare tutti i dati a utilizzare lo stesso sistema di riferimento spaziale e lo stesso sistema di coordinate.

Prerequisiti:

Per registrare una colonna spaziale, è necessario che l’ID utilizzato abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella a cui appartiene la colonna.
- Privilegio CONTROL o ALTER per la tabella.

Se per impostare i dati da un file di trasferimento SDE si utilizza il processore riga comandi db2se o un programma applicativo, è possibile impostare DB2 Spatial Extender in modo che crei e registri automaticamente la colonna in cui inserire i dati. In questo caso, è necessario che l’ID utente utilizzato disponga dell’autorizzazione SYSADM o DBADM per il database.

Procedura:

E’ possibile registrare una colonna spaziale nei seguenti modi:

- Utilizzando la finestra Colonne spaziali e la finestra di selezione del sistema di riferimento spaziale del Centro di controllo DB2.
- Con il comando **db2se register_spatial_column**.
- Eseguendo un’applicazione che richiama la procedura memorizzata db2gse.ST_register_spatial_column.
- Se si desidera importare dati spaziali da un file di trasferimento SDE, è possibile creare una tabella che contiene una colonna spaziale, registrare la colonna e importare i dati nella colonna, utilizzando la finestra Importazione dati spaziali del Centro di controllo, il comando **import_sde** oppure la procedura memorizzata db2gse.ST_import_sde.

Per controllare il sistema di riferimento spaziale scelto per una determinata colonna dopo averla registrata, fare riferimento alla colonna SRS_NAME nella vista DB2GSE.GSE_GEOMETRY_COLUMNS.

Impostazione delle colonne spaziali

Attività correlate:

- “Scrittura di applicazioni per DB2 Spatial Extender” a pagina 133

Riferimenti correlati:

- “Richiamo dei comandi per l’impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125
- “ST_register_spatial_column” a pagina 270

Capitolo 10. Inserimento dei dati nelle colonne spaziali

Dopo aver creato le colonne spaziali e registrato le colonne a cui dovranno accedere gli strumenti di visualizzazione, è possibile inserire i dati spaziali nelle colonne. E' possibile inserire i dati importandoli, utilizzando un geocoder in grado di ricavare i dati da dati aziendali oppure utilizzando funzioni spaziali che creano i dati o li estraggono da dati aziendali o da un altro tipo di dati spaziali. Il presente capitolo:

- Descrive le operazioni di importazione dei dati nel database spaziale e di esportazione dei dati in file utilizzabili dall'applicazione.
- Descrive la funzione di geocoding ed illustra le attività di impostazione delle operazioni di geocoding ed inoltre l'esecuzione automatica e in modalità batch dei geocoder.

Importazione ed esportazione dei dati spaziali

Questa sezione descrive le operazioni di importazione ed esportazione dei dati ed introduce le seguenti attività:

- Importazione dei dati spaziali in una nuova tabella, o in una tabella o vista esistenti
- Esportazione dei dati spaziali in file utilizzabili dall'applicazione

Informazioni sull'importazione e l'esportazione dei dati spaziali

E' possibile utilizzare DB2[®] Spatial Extender per scambiare i dati tra il database e origini dati esterne. Più precisamente, è possibile importare i dati da origini esterne trasferendoli nel database sotto forma di file, definiti *file di scambio dati*. Inoltre i dati possono essere esportati dal database in file di scambio dati da cui le origini esterne possono prelevarli. In questa sezione vengono descritti alcuni dei motivi per cui importare ed esportare i dati spaziali e i tipi di file di scambio dati supportati da DB2 Spatial Extender.

Perché importare ed esportare i dati spaziali:

Importando i dati spaziali, è possibile ottenere una grande quantità di informazioni spaziali già disponibili nell'azienda. Esportandoli in un formato file standard diventeranno disponibili per le applicazioni esistenti. Considerare i seguenti esempi:

- Il database contiene dati spaziali che rappresentano gli uffici, i clienti e altri elementi aziendali. Si desidera supportare questi dati con dati spaziali che rappresentino l'ambiente culturale dell'azienda—città, strade, località di interesse e così via. I dati desiderati sono reperibili da un fornitore di carte geografiche. E' possibile utilizzare DB2 Spatial Extender per importare questi dati da un file di scambio dati del fornitore.
- Si desidera migrare i dati spaziali da un sistema Oracle all'ambiente DB2. Si procede utilizzando un'utilità Oracle che consente di scrivere i dati in un file di scambio dati. A questo punto, utilizzando DB2 Spatial Extender, importare i dati dal file di scambio dati nel database abilitato per le operazioni spaziali.

Inserimento dei dati nelle colonne spaziali

- L'utente non è collegato a DB2 e desidera utilizzare un geobrowser per mostrare le presentazioni delle informazioni spaziali ai clienti. Il browser necessita solo dei file da cui prelevare i dati; non è necessario stabilire la connessione a un database. E' possibile utilizzare DB2 Spatial Extender per esportare i dati in un file di scambio dati, quindi visualizzarli con un browser.

File shape e file di trasferimento SDE:

DB2 Spatial Extender supporta due tipi di file di scambio dati: file shape e file di trasferimento SDE. Il termine *file shape* si riferisce ad un insieme di file aventi stesso nome ma estensioni diverse. L'insieme può contenere fino a quattro file che sono:

- File che contengono dati spaziali in *formato shp*, sviluppato dalla ESRI. Questi dati vengono definiti generalmente *dati shape*. L'estensione di un file contenente dati shape è .shp.
- File che contengono dati aziendali appartenenti alle ubicazioni definite dai dati shape. L'estensione di questo tipo di file è .dbf.
- File che contengono un indice dei dati shape. L'estensione di questo tipo di file è .shx.
- File che contengono le specifiche del sistema di coordinate su cui si basano i dati contenuti in un file .shp. L'estensione di questo tipo di file è .prj.

I file shape vengono utilizzati per importare dati originati in file system e per esportare dati in file contenuti nei file system.

Se si utilizza DB2 Spatial Extender per importare i dati shape, verrà ricevuto almeno un file .shp. Nella maggior parte dei casi, si riceverà uno o più file appartenenti agli altri tipi indicati.

I *file di trasferimento SDE* vengono utilizzati per importare i dati originati nei database ESRI. Ciascun file comprende dei dati spaziali, un sistema di riferimento spaziale per questi dati, e dati aziendali. I dati spaziali, il cui formato appartiene a ESRI, vengono utilizzati nella colonna di una tabella registrata nel catalogo DB2 Spatial Extender. I dati aziendali sono destinati alle altre colonne della tabella a cui appartiene la colonna registrata.

Importazione di dati spaziali

Questa sezione contiene una panoramica delle attività di importazione dei dati shape e di trasferimento SDE nel database. Inoltre contiene dei riferimenti ad alcune specifiche necessarie per l'esecuzione delle attività (ad esempio, processi e parametri).

Importazione dei dati shape in una tabella nuova o esistente

I dati shape possono essere importati in una tabella esistente, oppure è possibile creare una nuova tabella e importarvi i dati in una singola operazione. Le operazioni che è possibile eseguire sono:

- Importare i dati shape nella colonna spaziale di una tabella esistente, in una vista esistente aggiornabile, oppure in una vista esistente in cui è definito un trigger INSTEAD OF per INSERT.
- Creare una tabella con una colonna spaziale e importarvi automaticamente i dati shape.

Prerequisiti:

Per importare i dati shape in una vista o tabella esistente, è necessario che l'ID utilizzato per eseguire l'operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella o la vista
- Privilegio CONTROL per la tabella o la vista
- Privilegio INSERT per la tabella o la vista
- Privilegio SELECT per la tabella o la vista (obbligatorio solo se la tabella contiene una colonna ID non di tipo IDENTITY)
- Privilegi di accesso alle directory a cui appartengono i file di input e i file di errore
- Privilegi di accesso in lettura ai file di input e in scrittura ai file di errore

Per creare una tabella in cui importare automaticamente i dati shape, è necessario che l'ID utilizzato per eseguire l'operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM, DBADM o CREATETAB per il database che contiene la tabella
- Uno dei seguenti privilegi:
 - Privilegio CREATEIN per lo schema a cui appartiene la tabella (obbligatorio se lo schema esiste già)
 - Autorizzazione IMPLICIT_SCHEMA per il database che contiene la tabella (obbligatoria se lo schema indicato per la tabella non è stato ancora creato)
- Privilegi di accesso alle directory a cui appartengono i file di input e i file di errore
- Privilegi di accesso in lettura ai file di input e in scrittura ai file di errore

Procedura:

E' possibile importare i dati shape nei seguenti modi:

- Utilizzando la finestra di importazione dei dati shape del Centro di controllo DB2.
- Con il comando **db2se import_shape**.
- Eseguendo un'applicazione che richiami la procedura memorizzata `db2gse.ST_import_shape`.

Suggerimenti:

E' possibile migliorare le prestazioni del processo di importazione, usufruendo delle funzioni disponibili in DB2. Ad esempio, durante l'importazione dei dati in una tabella esistente o in una nuova tabella, definire la tabella come NON COLLEGATA INIZIALMENTE specificando i parametri di creazione tabella appropriati.

Argomenti correlati:

- "Informazioni sull'importazione e l'esportazione dei dati spaziali" a pagina 87

Attività correlate:

- "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90
- "Scrittura di applicazioni per DB2 Spatial Extender" a pagina 133

Inserimento dei dati nelle colonne spaziali

Riferimenti correlati:

- “Richiamo dei comandi per l’impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125
- “ST_import_shape” a pagina 257

Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente

I dati di trasferimento SDE possono essere importati in una tabella esistente, oppure è possibile creare una nuova tabella e importarvi i dati in una singola operazione. Le operazioni che è possibile eseguire sono:

- Importare i dati di trasferimento SDE in una tabella esistente che comprende una colonna spaziale già registrata nel catalogo di DB2 Spatial Extender. I dati di trasferimento possono comprendere i dati spaziali per la colonna e i dati aziendali delle altre colonne nella tabella.
- Creare automaticamente una tabella che abbia una colonna spaziale, registrare la colonna nel catalogo e importare i dati di trasferimento SDE in questa colonna e nelle altre colonne della tabella.

Prerequisiti:

Per importare i dati in una colonna di una vista o tabella esistente, è necessario che l’ID utilizzato per eseguire l’operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella o la vista
- Privilegio CONTROL per la tabella o la vista
- Privilegi INSERT e SELECT per la tabella o vista

Per creare una tabella in cui importare automaticamente i dati shape, è necessario che l’ID utilizzato per eseguire l’operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM, DBADM o CREATETAB per il database che contiene la tabella
- Uno dei seguenti privilegi:
 - Privilegio CREATEIN per lo schema a cui appartiene la tabella (obbligatorio se lo schema esiste già)
 - Autorizzazione IMPLICIT_SCHEMA per il database che contiene la tabella (obbligatoria se lo schema indicato per la tabella non è stato ancora creato)

Procedura:

E’ possibile importare i dati di trasferimento SDE nei seguenti modi:

- Utilizzando la finestra di importazione del Centro di controllo DB2.
- Con il comando **db2se import_sde**.
- Eseguendo un’applicazione che richiami la procedura memorizzata `db2gse.GSE_import_sde`.

Per ulteriori informazioni sullo svolgimento di tali attività, consultare le “Attività correlate” riportate al termine di questa sezione.

Argomenti correlati:

- “Informazioni sull’importazione e l’esportazione dei dati spaziali” a pagina 87

Attività correlate:

- “Importazione dei dati shape in una tabella nuova o esistente” a pagina 88
- “Scrittura di applicazioni per DB2 Spatial Extender” a pagina 133

Riferimenti correlati:

- “Richiamo dei comandi per l’impostazione di DB2 Spatial Extender e per lo sviluppo di progetti” a pagina 125
- “GSE_import_sde” a pagina 228

Esportazione dei dati spaziali

Questa sezione contiene una panoramica delle attività di esportazione dei dati shape e di trasferimento SDE. Inoltre contiene dei riferimenti ad alcune specifiche necessarie per l’esecuzione delle attività (ad esempio, processi e parametri).

Esportazione dei dati in un file shape

E’ possibile esportare i dati spaziali restituiti da una interrogazione in un file shape. I dati possono provenire da diverse origini, quali una tabella di base, da più tabelle unite tra loro, da una serie di risultati di una interrogazione o dall’output di una funzione spaziale.

DB2 Spatial Extender può aggiungere i dati che si desidera esportare a un file esistente, oppure può creare un nuovo file.

Prerequisiti:

Per esportare i dati in un file shape, è necessario che l’ID utilizzato abbia uno dei seguenti privilegi:

- Privilegio di esecuzione di una selezione secondaria che restituisca i risultati che si desidera esportare
- Privilegio di scrittura nella directory in cui si trova il file che contiene i dati che si desidera esportare
- Privilegio di creazione di un file che contenga i dati esportati (richiesto se il file non esiste già)

Per ulteriori informazioni sui privilegi e sulle relative modalità di assegnazione, rivolgersi al responsabile del database.

Procedura:

E’ possibile esportare i dati in un file shape nei seguenti modi:

- Avviando l’esportazione dalla finestra di esportazione dei file shape del Centro di controllo DB2.
- Utilizzando il comando **db2se export_shape** dal processore riga comandi db2se.
- Eseguendo un’applicazione che richiami la procedura memorizzata db2gse.ST_export_shape.

Per ulteriori informazioni sullo svolgimento di tali attività, consultare le “Attività correlate” riportate al termine di questa sezione.

Esportazione dei dati in un file di trasferimento SDE

E' possibile esportare una tabella che contiene dati spaziali in un file di trasferimento SDE. La tabella non potrà contenere più di una colonna spaziale. Inoltre, la colonna dovrà essere registrata nel catalogo di DB2 Spatial Extender. Se la tabella contiene dati aziendali, questi verranno esportati insieme ai dati spaziali. E' possibile esportare tutte le righe della tabella o solo una serie di righe. Per esportare una serie di righe, indicare una clausola WHERE che identifichi la serie.

Prerequisiti:

Per esportare i dati in un file di trasferimento SDE, è necessario che l'ID utilizzato abbia una delle seguenti autorizzazioni:

- Autorizzazione SYSADM o DBADM.
- Privilegio SELECT per la tabella da esportare.
- Privilegio di scrittura nella directory in cui si trova il file che contiene i dati che si desidera esportare

Limitazioni:

- E' possibile esportare solo una colonna spaziale alla volta.
- Le colonne esportate devono contenere dati supportati dal formato SDE.
- La tabella potrà contenere solo una colonna spaziale.
- La colonna dovrà essere registrata nel catalogo di DB2 Spatial Extender.
- Non è possibile utilizzare file SDE esistenti.

Procedura:

E' possibile esportare i dati spaziali e i dati aziendali in un file di trasferimento SDE in uno dei seguenti modi:

- Utilizzando la finestra di esportazione dei file SDE del Centro di controllo DB2.
- Con il comando **db2se export_sde**.
- Eseguendo un'applicazione che richiami la procedura memorizzata `db2gse.GSE_export_sde`.

Argomenti correlati:

- "Informazioni sull'importazione e l'esportazione dei dati spaziali" a pagina 87

Attività correlate:

- "Scrittura di applicazioni per DB2 Spatial Extender" a pagina 133

Riferimenti correlati:

- "Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti" a pagina 125
- "GSE_export_sde" a pagina 226

Uso del geocoder

Questa sezione descrive le funzioni di geocoding ed introduce le seguenti attività:

- Definizione delle operazioni che dovranno essere eseguite dal geocoder; ad esempio, specificare il numero di record da elaborare prima di eseguire un commit

- Impostazione del geocoder affinché esegua il geocoding dei dati nel momento in cui i dati vengono aggiunti o aggiornati in una tabella.
- Esecuzione del geocoder in modalità batch

Geocoder e funzioni di geocoding

I termini *geocoder* e *geocoding* sono utilizzati in diversi contesti nel corso di questa pubblicazione. Questo paragrafo contiene la definizione di *geocoder* e *geocoding*, definisce le modalità in cui un geocoder funziona, descrive il processo globale a cui appartengono le funzioni di geocoding e riepiloga le attività dell'utente relative alle funzioni di geocoding.

In DB2[®] Spatial Extender, un geocoder è una funzione scalare che converte dei dati esistenti (l'input della funzione) in dati concreti se riferiti allo spazio (l'output della funzione). Generalmente, i dati esistenti sono dati relazionali che descrivono una posizione. Ad esempio, il geocoder fornito con DB2 Spatial Extender, DB2SE_USA_GEOCODER, converte gli indirizzi degli Stati Uniti in dati ST_Point. DB2 Spatial Extender supporta altri geocoder di utenti o di fornitori esterni, i cui input e output non dovranno essere necessariamente uguali a quelli di DB2SE_USA_GEOCODER. Ad esempio, il geocoder ottenuto da un fornitore potrebbe convertire indirizzi in coordinate che non verranno memorizzate da DB2, ma bensì scritte in un file, mentre un altro geocoder, potrebbe essere in grado di convertire i numeri di un ufficio situato in un edificio commerciale in coordinate che indicano la posizione dell'ufficio nell'edificio, oppure convertire il numero di riferimento di uno scaffale di un magazzino in coordinate che indicano la posizione di quello scaffale nel magazzino.

In altri casi, i dati esistenti convertiti dal geocoder sono dati spaziali. Ad esempio, un geocoder fornito dall'utente potrebbe convertire le coordinate X e Y in dati conformi ai tipi di dati utilizzati da DB2 Spatial Extender.

In DB2 Spatial Extender, la funzione di *geocoding* è semplicemente l'operazione eseguita dal geocoder per convertire gli input in output, ad esempio convertire indirizzi in coordinate.

Modalità:

Un geocoder opera in due modalità:

- In *modalità batch*, il geocoder, in una sola operazione, tenta di convertire tutti gli input di una singola tabella. Ad esempio, in modalità batch, il geocoder DB2SE_USA_GEOCODER converte tutti gli indirizzi presenti in una tabella (oppure tutti gli indirizzi di una serie di righe della tabella).
- In *modalità automatica*, il geocoder converte i dati nel momento in cui vengono inseriti o aggiornati in una tabella. Il geocoder viene attivato dai trigger INSERT e UPDATE definiti nella tabella.

Funzioni di geocoding:

La funzione di geocoding è una delle operazioni che consente di ottenere il contenuto di una colonna spaziale di una tabella DB2 da altri dati. In questo paragrafo tali operazioni vengono definite complessivamente *funzione di geocoding*. Le funzioni di geocoding variano da geocoder in geocoder. Ad esempio, DB2SE_USA_GEOCODER, per determinare se l'indirizzo specificato come input corrisponde ad un indirizzo noto in base al livello di corrispondenza indicato, esegue una ricerca in file contenenti indirizzi noti. Gli indirizzi noti, a cui gli utenti

Inserimento dei dati nelle colonne spaziali

fanno riferimento nel corso delle ricerche, vengono definiti *dati di riferimento*. Non tutti i geocoder si basano su dati di riferimento. Di seguito sono riportate le funzioni di geocoding di DB2SE_USA_GEOCODER:

1. DB2SE_USA_GEOCODER esegue le operazioni per cui è stato ideato:
 - a. DB2SE_USA_GEOCODER analizza ciascun indirizzo che riceve come input.
 - b. DB2SE_USA_GEOCODER ricerca i dati di riferimento relativi ai nomi di strade che assomigliano agli indirizzi analizzati e limita la ricerca alle strade che si trovano nell'area indicata dal codice postale dell'indirizzo.
 - c. Se la ricerca ha esito positivo, DB2SE_USA_GEOCODER stabilisce quale indirizzo, tra le strade individuate, corrisponde all'indirizzo analizzato.
 - d. Se DB2SE_USA_GEOCODER rileva una corrispondenza, esegue il geocoding dell'indirizzo, in caso contrario non restituisce alcun risultato.
2. Quando DB2SE_USA_GEOCODER esegue il geocoding dell'indirizzo, DB2 inserisce le coordinate ottenute nella colonna spaziale specificata.
3. Quando DB2SE_USA_GEOCODER esegue il geocoding in modalità batch, DB2 Spatial Extender esegue il commit dei dati (a) ogni qual volta DB2SE_USA_GEOCODER completa l'elaborazione di un certo numero di record o (b) quando DB2SE_USA_GEOCODER completa l'elaborazione di tutti gli input.

Attività dell'utente:

In DB2 Spatial Extender, le attività relative alla funzione di geocoding sono:

- Indicare in che modo eseguire determinate parti del processo di geocoding per una specifica colonna spaziale; ad esempio, impostare il livello minimo in base al quale i nomi delle strade o gli indirizzi dei record di input corrispondono ai nomi delle strade o agli indirizzi dati di riferimento, e stabilire quanti record elaborare prima di eseguire il commit dei nuovi dati. Questa attività viene definita *impostazione delle funzioni di geocoding*.
- Specificare se eseguire il geocoding dei dati ogni qual volta vengono inseriti o aggiornati in una tabella. La funzione di geocoding automatica, fa sì che vengano eseguite le istruzioni specificate dall'utente durante l'impostazione delle operazioni (ad eccezione delle istruzioni che prevedono l'esecuzione del commit, che vengono applicate solo per l'esecuzione del geocoding in modalità batch). Questa attività viene definita *impostazione dell'esecuzione automatica del geocoder*.
- Eseguire il geocoder in modalità batch. Se l'utente ha già impostato le funzioni di geocoding, le istruzioni specificate verranno applicate in tutte le sessioni batch, a meno che non vengano sostituite da altre istruzioni. Se le funzioni di geocoding non sono ancora state impostate, è possibile specificare che le nuove impostazioni abbiano effetto solo per una particolare sessione. Questa attività viene definita *esecuzione del geocoder in modalità batch*.

Impostazione delle funzioni di geocoding

DB2 Spatial Extender consente di impostare in anticipo le attività da eseguire quando viene richiamato un geocoder. Ad esempio, è possibile specificare:

- La colonna a cui il geocoder dovrà fornire i dati.
- Se l'input letto dal geocoder nella tabella o vista deve essere limitato a una serie definita di righe.
- Il numero di record per cui eseguire il geocoding nelle sessioni batch in un'unità di lavoro.

- I requisiti per le operazioni di un geocoder specifico. Ad esempio, DB2SE_USA_GEOCODER può eseguire il geocoding solo dei record che corrispondono alle informazioni contenute nei dati di riferimento in base al livello di corrispondenza indicato. Questo livello di corrispondenza viene definito *valore minimo di corrispondenza*.

E' *necessario* specificare i parametri descritti prima di impostare il geocoder per l'esecuzione automatica. A questo punto, ad ogni avvio del geocoder (non solo in modalità automatica, ma anche in modalità batch), le funzioni di geocoding verranno eseguite in base alle impostazioni indicate. Ad esempio, se le impostazioni prevedono l'esecuzione del geocoding di 45 record in modalità batch in ciascuna unità di lavoro, viene eseguito un commit dopo ogni quarantacinquesimo record sottoposto a geocoding (eccezione: è possibile sostituire le specifiche delle singole sessioni di geocoding in modalità batch).

Non è necessario indicare dei valori predefiniti per le operazioni di geocoding prima di eseguire il geocoder in modalità batch. Piuttosto, all'avvio della sessione batch, è possibile indicare la modalità di esecuzione delle operazioni in base alla durata dell'esecuzione. Se vengono indicati dei valori predefiniti per le sessioni batch, è possibile modificarli per ogni singola sessione.

Prerequisiti:

Per impostare le funzioni di geocoding per un determinato geocoder, è necessario che l'ID utilizzato per eseguire l'operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM o DBADM per il database che contiene le tabelle che verranno analizzate dal geocoder.
- Privilegio SELECT e privilegio CONTROL o UPDATE per ciascuna tabella che verrà analizzata dal geocoder

Procedura:

E' possibile impostare le funzioni di geocoding nei seguenti modi:

- Richiamando la finestra di impostazione delle funzioni di geocoding del Centro di controllo DB2.
- Con il comando **db2se setup_gc**.
- Eseguendo un'applicazione che richiami la procedura memorizzata db2gse.ST_setup_geocoding.

Per ulteriori informazioni sullo svolgimento di tali attività, consultare le "Attività correlate" riportate al termine di questa sezione.

Suggerimenti:

- Quando DB2SE_USA_GEOCODER legge un record di dati di indirizzo, tenta di associare quel record ad un record corrispondente nei dati di riferimento. Il geocoder procede come segue: ricerca nei dati di riferimento nomi di strade con lo stesso codice postale del record fornito. Se trova un nome simile al nome indicato dal record rispettando il livello minimo di corrispondenza, da inizio alla ricerca dell'indirizzo completo. Se individua un indirizzo completo che corrisponde interamente al record indicato, esegue il geocoding del record. Se non individua alcun indirizzo, non restituisce alcun risultato.

Il valore minimo di corrispondenza in base al quale viene stabilita la corrispondenza dei nomi delle strade, è definito *corrispondenza ortografica*, mentre

Inserimento dei dati nelle colonne spaziali

il valore minimo di corrispondenza dell'indirizzo completo è definito *livello minimo di corrispondenza*. Ad esempio, se la corrispondenza ortografica è pari all'80%, affinché il geocoder dia inizio alla ricerca dell'indirizzo completo, è necessario che i nomi delle strade corrispondano almeno all'80%. Se il livello minimo di corrispondenza è 60%, affinché venga eseguito il geocoding del record, gli indirizzi devono avere una corrispondenza pari almeno al 60%.

E' possibile specificare il valore della corrispondenza ortografica e del livello minimo di corrispondenza. Potrebbe inoltre essere necessario ottimizzare i valori durante la ricerca. Ad esempio, se la corrispondenza ortografica e il livello minimo di corrispondenza siano pari al 95% e gli indirizzi che si desidera sottoporre a geocoding non sono stati precedentemente verificati, le possibilità che vi sia una corrispondenza del 95% sono pressoché inesistenti. Di conseguenza, il geocoder potrebbe non restituire alcun risultato durante l'elaborazione dei record. In questo caso, è preferibile diminuire il valore della corrispondenza ortografica ed eseguire nuovamente il geocoder. I valori consigliati per la corrispondenza ortografica e il livello minimo di corrispondenza, sono rispettivamente, 70 e 60

- Come riportato nella parte iniziale di questa sezione, è possibile stabilire se l'input letto dal geocoder nella tabella o vista deve essere limitato a una serie definita di righe. Ad esempio, fare riferimento ai seguenti scenari:
 - Si desidera eseguire il geocoding degli indirizzi in una tabella in modalità batch. Il livello minimo di corrispondenza è troppo elevato, quindi il geocoder non restituisce alcun risultato per la maggior parte di indirizzi analizzati. L'utente riduce il livello di corrispondenza ed esegue nuovamente il geocoder. Per far sì che l'input sia costituito solo dagli indirizzi non sottoposti a geocoding, è possibile indicare solo le righe per le quali non è stato restituito alcun valore nel corso dell'elaborazione precedente.
 - E' possibile indicare al geocoder solo le righe aggiunte dopo una certa data.
 - E' possibile indicare al geocoder solo le righe che contengono gli indirizzi di una particolare area; ad esempio un gruppo di regioni o uno stato.
- E' possibile anche indicare al geocoder il numero di record da elaborare nelle sessioni batch in un'unità di lavoro. Il geocoder può elaborare lo stesso numero di record in ciascuna unità di lavoro oppure elaborare tutti i record di una tabella in una singola unità. Nel secondo caso, tenere presente che:
 - Il controllo sulla dimensione dell'unità di lavoro è minore che nel primo caso. Di conseguenza, non è possibile controllare il numero di blocchi formati e il numero voci di registrazione inserite durante l'attività del geocoder.
 - Se il geocoder rileva un errore che necessita di un rollback, sarà necessario eseguire il geocoder nuovamente su tutti i record. Il costo di risorse che ne deriva può essere alto se la tabella è molto grande e se l'errore e il conseguente rollback si verificano quando la maggior parte dei record è già stata elaborata.

Impostazione dell'esecuzione automatica del geocoder

E' possibile impostare un geocoder in modo che converta automaticamente i dati nel momento in cui vengono aggiunti o aggiornati in una tabella.

Prerequisiti:

Per impostare un geocoder affinché venga eseguito automaticamente, procedere come segue:

- Impostare le funzioni di geocoding per ciascuna colonna spaziale in cui verranno inseriti i dati di output del geocoder.
- L'ID utente utilizzato dovrà avere una delle seguenti forme di autorizzazione:
 - Autorizzazione SYSADM o DBADM per il database che contiene la tabella in cui verranno definiti i trigger che richiameranno il geocoder
 - Uno o più privilegi per la tabella:
 - Privilegio CONTROL.
 - Se non si dispone del privilegio CONTROL, sono necessari i privilegi ALTER, SELECT e UPDATE.
 - I privilegi necessari per creare i trigger nella tabella.

Procedura:

E' possibile impostare la funzione di geocoding automatica in tre modi differenti:

- Richiamando la finestra di impostazione delle funzioni di geocoding del Centro di controllo DB2.
- Con il comando **db2se enable_autogc**.
- Eseguendo un'applicazione che richiami la procedura memorizzata `db2gse.ST_enable_autogeocoding`.

Per ulteriori informazioni sullo svolgimento di tali attività, consultare le "Attività correlate" riportate al termine di questa sezione.

Suggerimenti:

- E' possibile impostare l'esecuzione automatica del geocoder prima di avviarlo in modalità batch. In tal modo, l'esecuzione automatica verrà eseguita prima dell'esecuzione in modalità batch. In questo caso, è possibile che durante l'esecuzione del geocoder in modalità batch vengano elaborati i dati già elaborati durante l'esecuzione automatica. Come risultato non si avranno dati duplicati, perché i dati spaziali prodotti per la seconda volta, sostituiranno i dati precedenti. Tuttavia, questo processo può ridurre il livello delle prestazioni.
- Prima di stabilire se eseguire il geocoder in modalità automatica o in modalità batch, tenere presente che:
 - Le prestazioni risultano migliori durante l'esecuzione del geocoder in modalità batch. Una sessione batch viene avviata con una inizializzazione e si conclude con la cancellazione dei dati. Durante una sessione automatica, i dati vengono sottoposti a geocoding in una singola operazione, che comincia con l'inizializzazione e si conclude con la cancellazione dei dati.
 - Complessivamente, una colonna spaziale in cui i dati vengono inseriti mediante una funzione di geocoding in modalità batch, risulterà più aggiornata. Al termine di una sessione batch, i dati di indirizzo possono rimanere accumulati e non sottoposti a geocoding fino alla sessione successiva. Se invece, la funzione di geocoding automatica è già abilitata, i dati di indirizzo vengono sottoposti a geocoding non appena vengono memorizzati nel database.

Esecuzione del geocoder in modalità batch

E' possibile impostare il geocoder affinché venga eseguito in modalità batch; in tal modo vengono convertiti più record in dati spaziali da inserire in una determinata colonna in una singola operazione.

Inserimento dei dati nelle colonne spaziali

Ogni qual volta si desidera eseguire un geocoder per inserire i dati in una determinata colonna spaziale, è possibile impostare le funzioni di geocoding per quella colonna. L'impostazione delle funzioni prevede l'indicazione dei requisiti da rispettare durante l'esecuzione del geocoder. Ad esempio, se si desidera che DB2 Spatial Extender esegua un commit ogni 100 record di input elaborati dal geocoder, durante l'impostazione delle funzioni, sarà necessario specificare 100 come numero obbligatorio.

Prima di eseguire il geocoder, è possibile sostituire tutti i valori specificati durante l'impostazione delle funzioni. I nuovi valori saranno validi per tutta la durata dell'esecuzione.

Se non si desidera impostare le operazioni, è necessario, ad ogni avvio del geocoder, indicare i requisiti a cui attenersi durante l'esecuzione.

Prerequisiti:

Per eseguire il geocoder in modalità batch, è necessario che l'ID utilizzato per eseguire l'operazione abbia una delle seguenti forme di autorizzazione:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella i cui dati verranno sottoposti a geocoding
- Privilegio CONTROL o UPDATE per questa tabella
-

Per specificare il numero di record da elaborare prima di ciascun commit, è necessario disporre del privilegio SELECT. Se si desidera indicare le clausole WHERE per limitare le righe che dovranno essere analizzate dal geocoder, è necessario disporre del privilegio SELECT per tutte le tabelle e viste a cui si fa riferimento nelle clausole. Rivolgersi al responsabile del database.

Limitazioni:

Procedura:

E' impostare l'esecuzione del geocoder in modalità batch nei seguenti modi:

- Richiamando la finestra di esecuzione delle funzioni di geocoding del Centro di controllo DB2.
- Con il comando **db2se run_gc**.
- Eseguendo un'applicazione che richiami la procedura memorizzata `db2gse.ST_run_geocoding`.

Capitolo 11. Uso degli indici e delle viste per accedere ai dati spaziali

Prima di interrogare le colonne spaziali, è possibile creare degli indici e delle viste per semplificare l'accesso alle colonne. Il presente capitolo:

- Descrive la natura degli indici utilizzati da Spatial Extender per velocizzare l'accesso ai dati spaziali
- Descrive come creare tali indici
- Descrive come utilizzare le viste per accedere ai dati spaziali

Tipi di indici spaziali

Per ottenere migliori prestazioni nell'esecuzione delle interrogazioni, è importante che nella tabelle di base di un database siano definiti degli indici validi. Le prestazioni di un'interrogazione dipendono direttamente dalla velocità in cui è possibile rilevare i valori in una colonna durante l'interrogazione. Le interrogazioni che utilizzano gli indici vengono eseguite più velocemente e forniscono un miglioramento significativo delle prestazioni.

Le interrogazioni spaziali, generalmente, coinvolgono due o più dimensioni. Ad esempio, si consideri una interrogazione spaziale in cui si desidera sapere se un punto è incluso in un area (poligono). A causa della natura multidimensionale delle interrogazioni spaziali, la funzione di indicizzazione DB2® native B-tree non è sufficiente a risolvere questo tipo di interrogazione.

Le interrogazioni spaziali possono utilizzare i seguenti tipi di indice:

- Gli indici di griglia spaziale

La tecnologia di indicizzazione di DB2 Spatial Extender utilizza l'*indicizzazione di griglia*, progettata per l'indicizzazione dei dati spaziali multidimensionali e delle colonne spaziali. DB2 Spatial Extender fornisce un indice griglia ottimizzato per dati bidimensionali su una proiezione piana della Terra.

- Indici geodetici Voronoi

DB2 Geodetic Extender fornisce supporto per un nuovo metodo di accesso spaziale che consente di creare indici su colonne contenenti dati geodetici a più dimensioni. gli indici geodetici Voronoi sono più adatti ai dati geodetici rispetto agli indici di griglia perché considera la Terra come una sfera continua senza distorsioni vicino ai poli o al 180° meridiano.

Argomenti correlati:

- "Indici geodetici Voronoi" a pagina 175
- "Indici di griglia spaziali" a pagina 100

Attività correlate:

- "Creazione di indici geodetici Voronoi" a pagina 179
- "Creazione degli indici di griglia spaziali" a pagina 106

Riferimenti correlati:

- "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122

Indici di griglia spaziali

Gli indici migliorano le prestazioni delle interrogazioni alle applicazioni, soprattutto quando la tabella o le tabelle interrogate contengono molte righe. Se vengono creati gli indici appropriati che l'ottimizzatore delle interrogazioni sceglie per eseguire nuovamente l'interrogazione, il numero delle righe da elaborare, si riduce in modo consistente.

DB2 Spatial Extender fornisce un indice di griglia ottimizzato per due dati dimensionali. Tale indice viene creato sulle dimensioni X e Y di una forma geometrica.

Comprendere i seguenti aspetti di un indice di griglia può risultare utile:

- La generazione dell'indice
- L'uso delle funzioni spaziali in un'interrogazione
- Come un'interrogazione utilizza gli indici di griglia spaziali

Generazione degli indici di griglia spaziale

Spatial Extender genera gli indici di griglia spaziali utilizzando l'MBR (minimum bounding rectangle) di una forma geometrica. Nella maggior parte delle forme geometriche, l'MBR è un rettangolo che circonda la forma. Per ulteriori dettagli sugli MBR, consultare la sezione "ST_MBR" a pagina 429.

Un indice di griglia spaziale divide un'area in una griglia di quadrati logici con la dimensione fissa specificata durante la creazione dell'indice. L'indice spaziale viene costruito in base a una colonna spaziale, creando una o più voci per le intersezioni di ciascun MBR della forma geometrica con le celle della griglia. Le voci dell'indice consistono nell'identificativo della cella della griglia, l'MBR geometrico e l'identificativo interno della riga che contiene la forma geometrica.

E' possibile definire fino a tre livelli di indici spaziali (livelli di griglia). Utilizzando più livelli di griglia è possibile ottimizzare l'indice per le diverse dimensioni dei dati spaziali. Per ulteriori informazioni, consultare la sezione "Considerazioni sul numero di livelli di indice e dimensioni della griglia" a pagina 102.

Se una forma geometrica interseca quattro o più celle della griglia, viene passata al livello successivo. In generale, le forme geometriche più ampie verranno indicizzate ai livelli superiori. Se una forma geometrica interseca 10 o più celle in una griglia di massime dimensioni, viene utilizzato il livello di indice di overflow definito dal sistema. Questo livello di overflow previene la generazione di troppe voci di indice. Per migliorare le prestazioni, definire le dimensioni degli indici di griglia in modo da evitare che il livello di overflow venga utilizzato.

Ad esempio, se esistono più livelli di griglia, l'algoritmo di indicizzazione tenterà di utilizzare il livello di griglia inferiore in modo da presentare una risoluzione migliore dei dati indicizzati. Se una forma geometrica si interseca con più di quattro livelli di griglia, viene passata al livello successivo (ammesso che tale livello esista). Pertanto, se un indice spaziale presenta i tre livelli di griglia 10,0, 100,0 e 1000,0 intersecherà prima ciascuna forma geometrica con il livello di griglia 10,0. Le forme geometriche che intersecano quattro celle di griglia di dimensione 10,0, verranno intersecate con la griglia di livello 100,0. Se quattro o più intersezioni risultano al livello 100,0, la forma geometrica viene passata al livello 1000,0. Se più di 10 intersezioni risultano al livello 1000,0, la forma geometrica viene indicizzata nel livello di overflow.

Uso delle funzioni spaziali in un'interrogazione

L'ottimizzatore di DB2 UDB utilizza gli indici di griglia spaziali quando le interrogazioni contengono una delle seguenti funzioni nella clausola WHERE:

- ST_Contains
- ST_Crosses
- ST_Distance
- ST_EnvIntersects
- EnvelopesIntersect
- ST_Equals
- ST_Intersects
- ST_MBRIntersects
- ST_Overlaps
- ST_Touches
- ST_Within

Per ulteriori informazioni, consultare la sezione "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122.

Come un'interrogazione utilizza gli indici di griglia spaziali

Quando l'ottimizzatore delle interrogazioni sceglie un indice di griglia spaziale, l'esecuzione dell'interrogazione utilizza il seguente processo di filtraggio a più fasi:

1. Individua le celle di griglia che intersecano la finestra dell'interrogazione. Con *finestra dell'interrogazione* si intende la forma geometrica desiderata e specificata come secondo parametro nella funzione spaziale (fare riferimento agli esempi riportati di seguito).
2. Sottopone l'indice a scansione ricercando le voci con identificativi di celle della griglia corrispondenti.
3. Confronta i valori dell'MBR della forma geometrica nelle voci dell'indice con la finestra dell'interrogazione ed elimina qualsiasi valore esterno alla finestra.
4. Esegue altre analisi in base alle necessità. L'insieme di forme geometriche prescelto come risultato delle operazioni precedenti potrebbe essere sottoposto a ulteriori analisi per stabilire se soddisfano la funzione spaziale (ST_Contains, ST_Distance, ecc.). La funzione spaziale EnvelopesIntersect non esegue questa operazione e generalmente ottiene migliori prestazioni.

I seguenti esempi di interrogazioni spaziali hanno un indice di griglia spaziale nella colonna C.GEOMETRY:

```
SELECT name
FROM counties AS c
WHERE EnvelopesIntersect(c.geometry, -73.0, 42.0, -72.0, 43.0, 1) = 1
```

```
SELECT name
FROM counties AS c
WHERE ST_Intersects(c.geometry, :geometry2) = 1
```

Nel primo esempio, i quattro valori di coordinate definiscono la finestra dell'interrogazione. Questi valori di coordinate indicano gli angoli inferiore-sinistro e superiore-destro (42,0 -73,0 e 43,0 -72,0) di un rettangolo.

Nel secondo esempio, Spatial Extender elabora l'MBR della forma geometrica specificata dalla variabile host :geometry2 e lo utilizza come finestra dell'interrogazione.

Durante la creazione di un indice di griglia spaziale, specificare dimensioni per la griglia appropriate alle dimensioni della finestra di interrogazione più comune che verrà utilizzata dall'applicazione spaziale. Se la dimensione della griglia è elevata, le voci di indice delle forme geometriche esterne alla finestra dell'interrogazione dovranno essere sottoposte a scansione in quanto si trovano in celle della griglia che intersecano la finestra, e queste scansioni extra peggiorano le prestazioni. Tuttavia, una dimensione di griglia inferiore potrebbe generare più voci di indice per ciascuna forma geometrica che dovranno essere sottoposte a scansione, peggiorando comunque le prestazioni dell'interrogazione.

DB2 Spatial Extender fornisce il programma di utilità Index Advisor che analizza i dati delle colonne spaziali e suggerisce le dimensioni per la griglia appropriate alle dimensioni della finestra di interrogazione che verrà utilizzata più frequentemente. Per ulteriori informazioni, consultare la sezione "Determinazione delle dimensioni della griglia per un indice di griglia spaziale" a pagina 111.

Argomenti correlati:

- "Considerazioni sul numero di livelli di indice e dimensioni della griglia" a pagina 102
- "Tipi di indici spaziali" a pagina 99
- "Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica" a pagina 110

Attività correlate:

- "Creazione degli indici di griglia spaziali" a pagina 106

Riferimenti correlati:

- "Istruzione CREATE INDEX per l'indice di griglia spaziale" a pagina 108

Considerazioni sul numero di livelli di indice e dimensioni della griglia

Utilizzare Index Advisor per determinare le dimensioni di griglia appropriate per gli indici di griglia spaziali, in quanto si tratta del modo migliore per regolare gli indici e migliorare le prestazioni delle interrogazioni spaziali. Questa sezione contiene alcune spiegazioni relative agli effetti dei diversi livelli di griglia e dimensioni.

Numero dei livelli di griglia

E' possibile utilizzare un massimo di tre livelli. Tuttavia, per ciascun livello in un indice di griglia, durante le interrogazioni spaziali verrà eseguita una ricerca di indice di griglia separata per ciascun livello dell'indice. Quindi, se si dispone di più livelli di griglia, l'interrogazione risulta meno efficiente.

Se la dimensione dei valori di una colonna è relativamente costante, utilizzare un livello di griglia singolo. Tuttavia, una colonna spaziale tipica, non contiene forme geometriche della stessa dimensione relativa, ma le forme possono essere raggruppate nelle colonne in base alla dimensione. Si consiglia di far corrispondere i livelli di griglia ai gruppi di forme geometriche.

Ad esempio, se si dispone di una tabella di appezzamenti di terreno contenente una colonna spaziali con gruppi di piccoli appezzamenti urbani, circondati da appezzamenti rurali più estesi. Poiché le dimensioni degli appezzamenti possono essere raggruppate in due gruppi (piccoli urbani e grandi rurali), in questo caso è opportuno indicare due livelli di griglia per l'indice spaziale.

Dimensione delle celle della griglia

La regola generale consiste nel diminuire al massimo le dimensioni della griglia in modo da ottenere una risoluzione migliore riducendo il numero di voci di indice.

- E' preferibile utilizzare un valore basso per la dimensione più piccola della griglia in modo da ottimizzare l'intero indice per tutte le forme geometriche più piccole presenti nella colonna. In tal modo le forme geometriche che non rientrano nell'area di ricerca non verranno esaminate. Tuttavia, una dimensione molto piccola assegnata alla griglia produce anche un numero di voci di indice più elevato. Di conseguenza, il numero di voci dell'indice analizzate durante l'interrogazione aumenta, così come la quantità di spazio necessaria per memorizzare l'indice. Questi fattori riducono la qualità delle prestazioni.
- Utilizzando dimensioni più grandi per la griglia, l'indice viene ottimizzato per l'analisi di forme geometriche più ampie. Una dimensione più grande produce una minore quantità di voci di indice rispetto a una dimensione più piccola. Di conseguenza, l'indice richiederà una quantità di memoria inferiore e le prestazioni miglioreranno considerevolmente.

Le seguenti figure mostrano gli effetti delle diverse dimensioni della griglia.

La Figura 13 mostra una carta geografica di alcuni appezzamenti di terreno, ciascuno di essi rappresentato da un poligono. Il rettangolo nero rappresenta la finestra in cui viene eseguita l'interrogazione. Si supponga che si desidera trovare tutte le forme geometriche il cui MBR si interseca con la finestra di interrogazione. La Figura 13 mostra che 28 forme geometriche (evidenziate in rosa) hanno un MBR che si interseca con al finestra di interrogazione.

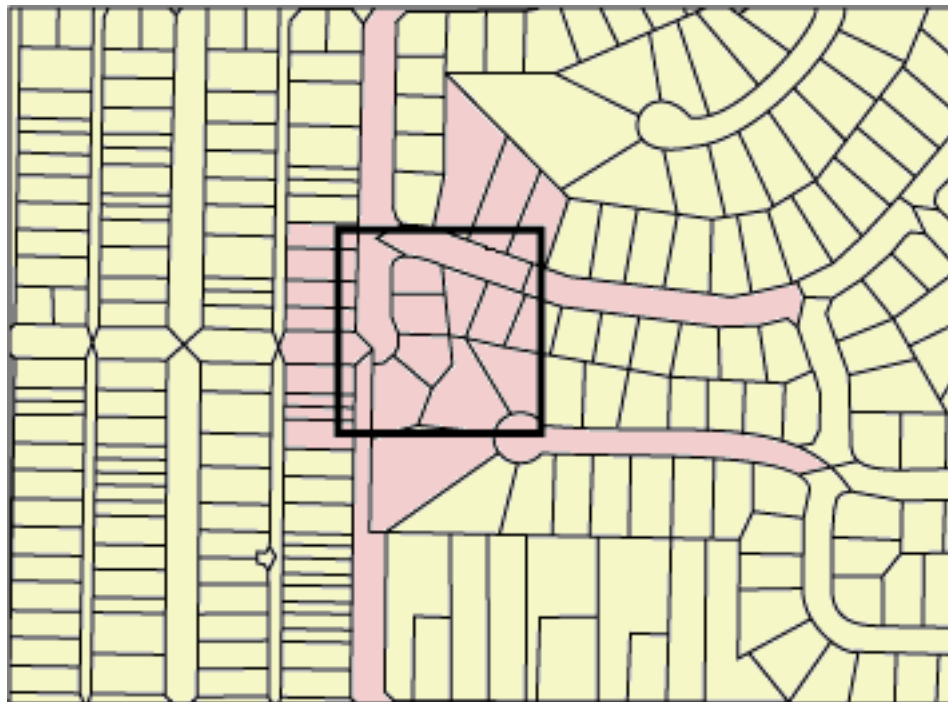


Figura 13. Appezzamenti di terreno

La Figura 14 a pagina 104 contiene una griglia di piccole dimensioni (25) che si avvicina di molto alla finestra di interrogazione.

Uso degli indici e delle viste

- L'interrogazione restituisce solo le 28 forme geometriche evidenziate, ma avrà esaminato ed eliminato altre tre forme geometriche i cui MBR si intersecano con la finestra.
- Questa griglia di piccole dimensioni genera molte voci di indice per ciascuna forma geometrica. Durante l'esecuzione, l'interrogazione accede a tutte le voci di indice delle 31 forme geometriche. La Figura 14 contiene 256 celle della griglia che si sovrappongono alla finestra dell'interrogazione. Tuttavia, l'interrogazione accede a 578 voci di indice, perché molte forme geometriche sono state indicizzate con le stesse celle della griglia.

Per questa finestra di interrogazione, la griglia di piccole dimensioni ottiene un numero eccessivo di voci di indice da esaminare.

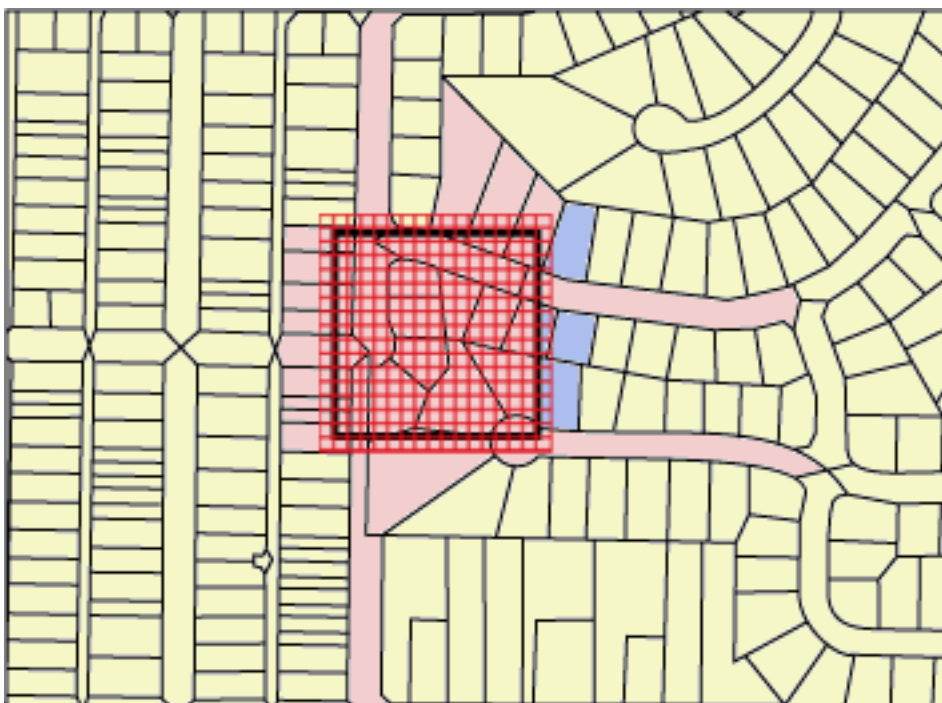


Figura 14. Griglia di piccole dimensioni (25) sugli appezzamenti di terreno

La Figura 15 a pagina 105 mostra una griglia di grandi dimensioni (400) che racchiude un'area decisamente più ampia contenente molte più forme geometriche della finestra di interrogazione.

- Questa griglia di grandi dimensioni ottiene solo una voce di indice per forma geometrica, ma l'interrogazione deve esaminare ed eliminare 59 forme geometriche i cui MBR si intersecano con la cella della griglia.
- Durante l'esecuzione, l'interrogazione accede a tutte le voci di indice per le 28 forme geometriche che si intersecano con la finestra di interrogazione, più le voci di indice per le 59 forme geometriche, per un totale di 112 voci di indice.

Per questa finestra di interrogazione, la griglia di grandi dimensioni ottiene un numero eccessivo di forme geometriche da esaminare.

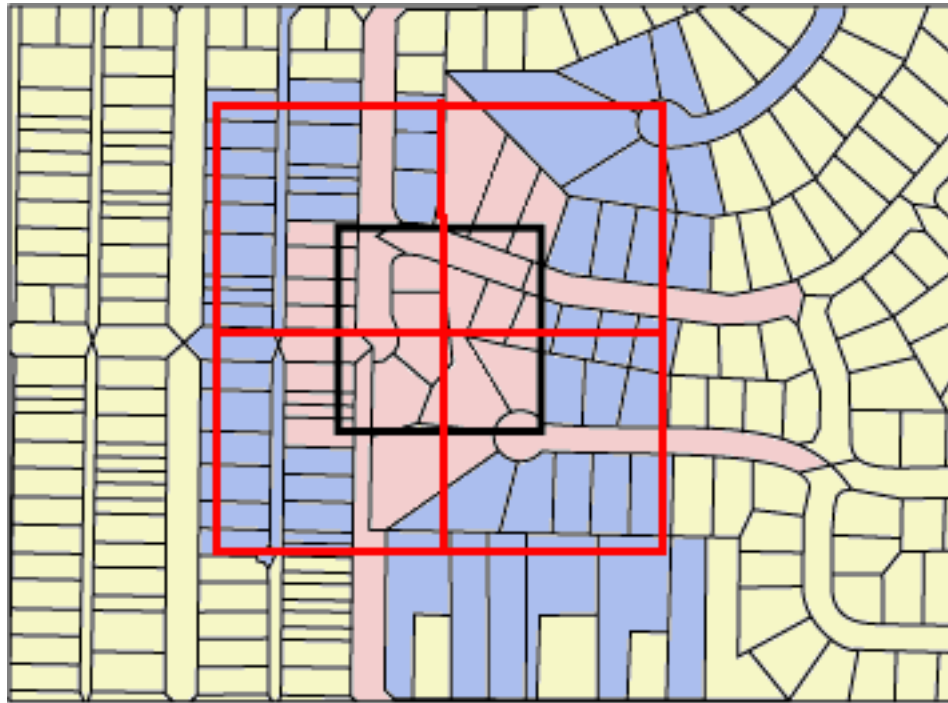


Figura 15. Griglia di grandi dimensioni (400) sugli appezzamenti di terreno

La Figura 16 a pagina 106 contiene una griglia di medie dimensioni (100) che si avvicina di molto alla finestra di interrogazione.

- L'interrogazione restituisce solo le 28 forme geometriche evidenziate, ma avrà esaminato ed eliminato altre tre cinque geometriche i cui MBR si intersecano con la finestra.
- Durante l'esecuzione, l'interrogazione accede a tutte le voci di indice per le 28 forme geometriche che si intersecano con la finestra di interrogazione, più le voci di indice per le 5 forme geometriche, per un totale di 91 voci di indice.

Per questa finestra di interrogazione, la griglia di medie dimensioni risulta essere la più adatta, in quanto ottiene un numero decisamente inferiore di voci di indice della griglia di piccole dimensioni e l'interrogazione può esaminare un numero inferiore di forme geometriche della griglia di grandi dimensioni.

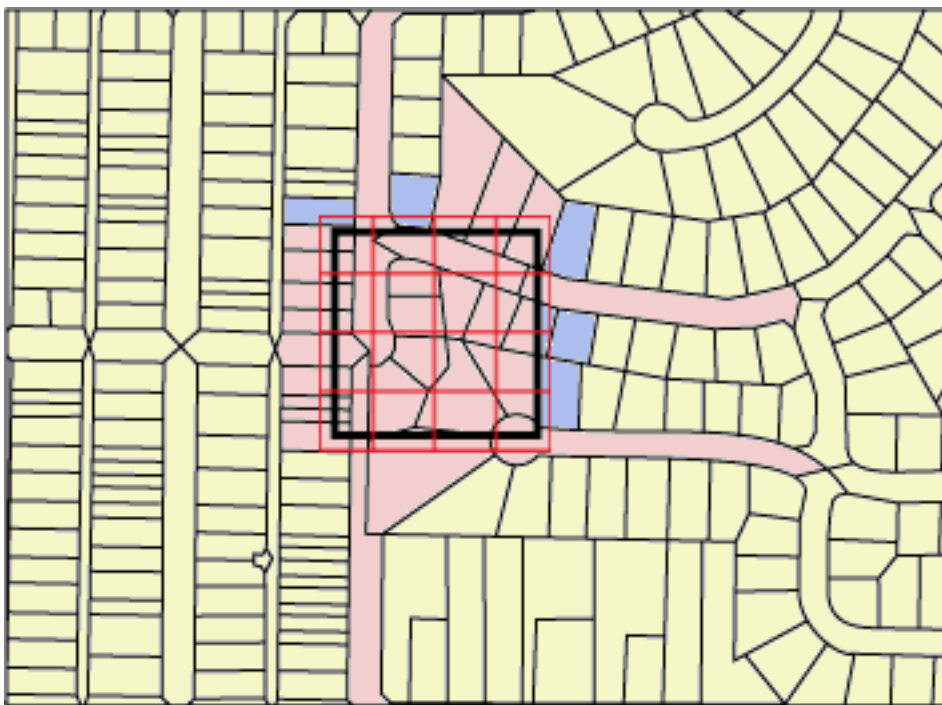


Figura 16. Griglia di medie dimensioni (100) sugli appezzamenti di terreno

Argomenti correlati:

- “Indici di griglia spaziali” a pagina 100
- “Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica” a pagina 110

Attività correlate:

- “Determinazione delle dimensioni della griglia per un indice di griglia spaziale” a pagina 111
- “Analisi delle statistiche dell’indice di griglia spaziale” a pagina 112
- “Creazione degli indici di griglia spaziali” a pagina 106

Riferimenti correlati:

- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

Creazione degli indici di griglia spaziali

Gli indici di griglia spaziali vengono creati per migliorare le prestazioni delle interrogazioni nelle colonne spaziali.

Durante la creazione dell’indice di griglia spaziale, verranno fornite le seguenti informazioni:

- Un nome
- Il nome della colonna spaziale su cui viene definito
- Le dimensioni della griglia (consultare la sezione “Indici di griglia spaziali” a pagina 100). La combinazione delle tre dimensioni della griglia consente di ottimizzare le prestazioni minimizzando il numero totale di voci di indice da sottoporre a scansione per eseguire l’interrogazione.

Prerequisiti:

Prima di creare un indice di griglia spaziale:

- L'ID utente utilizzato deve disporre delle autorizzazioni necessarie per l'istruzione DB2 SQL CREATE INDEX. L'ID utente deve disporre almeno di una delle seguenti autorizzazioni o privilegi:
 - Autorizzazione SYSADM o DBADM per il database a cui appartiene la tabella che contiene la colonna
 - Entrambi i seguenti privilegi o autorizzazioni:
 - Uno dei seguenti privilegi di tabella:
 - Privilegio CONTROL per la tabella
 - Privilegio INDEX per la tabella
 - Una delle seguenti autorizzazioni o privilegi per lo schema:
 - Autorizzazione IMPLICIT_SCHEMA per il database, se lo schema implicito o esplicito dell'indice non esiste
 - Privilegio CREATEIN per lo schema, se il nome dello schema dell'indice si riferisce a uno schema esistente
- E' necessario essere a conoscenza dei valori che si desidera specificare per il nome completo e per le dimensioni che l'indice utilizzerà. Per ulteriori informazioni, consultare la sezione "Determinazione delle dimensioni della griglia per un indice di griglia spaziale" a pagina 111.

Suggerimenti:

- Prima di creare un indice di griglia spaziale per una colonna, determinare i parametri necessari utilizzando Index Advisor. Index Advisor è in grado di analizzare i dati della colonna spaziale e di suggerire le dimensioni della griglia appropriate per l'indice di griglia spaziale.
- Se si desidera caricare i dati nella colonna, creare l'indice di griglia spaziale dopo aver completato il processo di caricamento. In tal modo, è possibile scegliere la dimensione delle celle più adatta in base alle caratteristiche dei dati utilizzando Index Advisor. Inoltre, caricando i dati prima dell'indice, le prestazioni del processo di caricamento migliorano sensibilmente in quanto non sarà necessario conservare l'indice durante l'operazione.

Limitazioni:

Durante la creazione di un indice di griglia spaziale, valgono le stesse limitazioni previste per la creazione degli indici mediante l'istruzione CREATE INDEX. ovvero la colonna in cui viene creato l'indice dovrà essere la colonna di una tabella di base non una colonna di viste o con nickname. DB2 UDB risolverà gli alias durante il processo.

Procedura:

E' possibile creare un indice di griglia spaziale in uno dei seguenti modi:

- Utilizzando la finestra Spatial Extender del Centro di controllo DB2.
- Utilizzando l'istruzione SQL CREATE INDEX con l'estensione db2gse.spatial_index nella clausola EXTEND USING.
- Utilizzando uno strumento GIS che funziona con DB2 Spatial Extender. Tale strumento utilizzerà l'istruzione SQL CREATE INDEX appropriata.

Uso degli indici e delle viste

La sezione contiene le operazioni da eseguire relative ai primi due metodi. Per informazioni sull'utilizzo di uno strumento GIS per la creazione di un indice di griglia spaziale, fare riferimento alla documentazione fornita con lo strumento.

Per creare un'indice di griglia spaziale utilizzando il Centro di controllo, procedere come segue:

1. Nel Centro di controllo, fare clic con il tasto destro del mouse sulla tabella della colonna spaziale per la quale si desidera creare l'indice, e selezionare **Spatial Extender** → **Indici spaziali** dal menu. Viene aperta la finestra Indici spaziali.
2. Attenersi alle istruzioni riportate nella guida in linea relativa alla finestra. Per visualizzare le istruzioni, scegliere il pulsante ? nella finestra Indici spaziali.

Per eseguire questa attività utilizzando l'istruzione SQL CREATE INDEX:

1. Determinare l'istruzione CREATE INDEX utilizzando la clausola EXTEND USING e l'estensione di indice di griglia db2gse.spatial_index.

Ad esempio, la seguente istruzione creerà l'indice di griglia spaziale TERRIDX per la tabella BRANCHES contenente la colonna spaziale TERRITORY.

```
CREATE INDEX terridx  
  ON branches (territory)  
  EXTEND USING db2gse.spatial_index (1.0, 10.0, 100.0)
```

2. Eseguire il comando CREATE nell'Editor comandi DB2, la finestra Comandi DB2 o il processore riga comandi DB2.

Argomenti correlati:

- "Indici di griglia spaziali" a pagina 100
- "Considerazioni sul numero di livelli di indice e dimensioni della griglia" a pagina 102
- "Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica" a pagina 110

Attività correlate:

- "Determinazione delle dimensioni della griglia per un indice di griglia spaziale" a pagina 111
- "Analisi delle statistiche dell'indice di griglia spaziale" a pagina 112

Riferimenti correlati:

- "CREATE INDEX statement" nella *SQL Reference, Volume 2*
- "Istruzione CREATE INDEX per l'indice di griglia spaziale" a pagina 108
- "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122

Istruzione CREATE INDEX per l'indice di griglia spaziale

Utilizzare l'istruzione CREATE INDEX con la clausola EXTEND USING per creare un indice di griglia spaziale.

Sintassi:

```
►► CREATE INDEX [schema_indice.] nome_indice ON →
```

```

▶ ┌───────────────────────────┐ nome_tabella ─ (─ nome_colonna ─) ─ EXTEND USING ────────────────────────────▶
  │ schema_tabella. │
  └───────────────────────────┘
▶ db2gse.spatial_index ─ (─ dimensione_minima_griglia ─, ────────────────────────────▶
▶ ─ dimensione_media_griglia ─, ─ dimensione_massima_griglia ─) ────────────────────────────▶

```

Parametri:

schema_indice.

Nome dello schema a cui appartiene l'indice creato. Se non viene specificato alcun nome, DB2 UDB utilizzerà il nome dello schema memorizzato nel registro speciale CURRENT SCHEMA.

nome_indice

Nome dell'indice di griglia creato.

schema_tabella.

Nome dello schema a cui appartiene la tabella che contiene la *nome_colonna*. Se non viene specificato alcun nome, DB2 utilizzerà il nome dello schema memorizzato nel registro speciale CURRENT SCHEMA.

nome_tabella

Nome della tabella che contiene la *nome_colonna*.

nome_colonna

Nome della colonna spaziale in cui viene creato l'indice di griglia.

dimensione_minima_griglia, dimensione_media_griglia, dimensione_massima_griglia

Dimensioni della griglia dell'indice spaziale. Questi parametri devono rispettare le seguenti condizioni:

- La *dimensione_minima_griglia* deve essere maggiore di 0.
- La *dimensione_media_griglia* deve essere maggiore alla *dimensione_minima_griglia* o essere 0.
- La *dimensione_massima_griglia* deve essere maggiore della *dimensione_media_griglia* o essere 0.

La validità delle dimensioni della griglia viene verificata alla prima indicizzazione della forma geometrica, sia che l'indice sia stato creato mediante l'istruzione CREATE INDEX che mediante il Centro di controllo. In tal modo, se le dimensioni della griglia indicate non soddisfano le impostazioni previste, viene generata una condizione di errore nelle seguenti situazioni:

- Se tutte le forme geometriche della colonna spaziale sono nulle, Spatial Extender creerà l'indice con esito positivo, senza verificare la validità delle dimensioni della griglia. Spatial Extender convaliderà le impostazioni della griglia quando viene inserita o aggiornata una forma geometrica non nulla nella colonna spaziale. Se le dimensioni della griglia specificate non sono valide, nel momento in cui la forma geometrica non nulla viene inserita o aggiornata, si verificherà un errore.
- Se durante la creazione dell'indice, la colonna spaziale contiene delle forme geometriche non nulle, Spatial Extender convaliderà le dimensioni della griglia. Se le dimensioni specificate non sono valide, si verifica immediatamente un errore e l'indice di griglia spaziale non viene creato.

Esempi:

Il seguente esempio di istruzione CREATE INDEX crea l'indice di griglia spaziale TERRIDX nella colonna spaziale TERRITORY nella tabella BRANCHES:

Uso degli indici e delle viste

```
CREATE INDEX terridx  
  ON branches (territory)  
  EXTEND USING db2gse.spatial_index (1.0, 10.0, 100.0)
```

Argomenti correlati:

- “Considerazioni sul numero di livelli di indice e dimensioni della griglia” a pagina 102
- “Indici di griglia spaziali” a pagina 100
- “Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica” a pagina 110

Attività correlate:

- “Determinazione delle dimensioni della griglia per un indice di griglia spaziale” a pagina 111
- “Analisi delle statistiche dell’indice di griglia spaziale” a pagina 112
- “Creazione degli indici di griglia spaziali” a pagina 106

Riferimenti correlati:

- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

Regolazione degli indici di griglia spaziali con Index Advisor

Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica

DB2[®] Spatial Extender è dotato del programma di utilità *Index Advisor*, che è possibile utilizzare per:

- Determinare dimensioni di griglia appropriate per gli indici di griglia spaziali. Index Advisor analizza le forme geometriche di una colonna spaziale e suggerisce le dimensioni di griglia ottimali per l’indice di griglia spaziale. Per informazioni su questa procedura, consultare la sezione “Determinazione delle dimensioni della griglia per un indice di griglia spaziale” a pagina 111.
- Analizzare un indice di griglia esistente. Index Advisor è in grado di raccogliere e visualizzare le statistiche da cui sarà possibile determinare come le dimensioni delle celle della griglia corrente faciliteranno il richiamo dei dati spaziali. Per informazioni su questa procedura, consultare la sezione “Analisi delle statistiche dell’indice di griglia spaziale” a pagina 112.

Argomenti correlati:

- “Considerazioni sul numero di livelli di indice e dimensioni della griglia” a pagina 102
- “Indici di griglia spaziali” a pagina 100

Attività correlate:

- “Determinazione delle dimensioni della griglia per un indice di griglia spaziale” a pagina 111
- “Analisi delle statistiche dell’indice di griglia spaziale” a pagina 112

Riferimenti correlati:

- “Istruzione CREATE INDEX per l’indice di griglia spaziale” a pagina 108
- “Comando gseidx” a pagina 117
- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

Determinazione delle dimensioni della griglia per un indice di griglia spaziale

Prima di creare un indice di griglia spaziale per una colonna, è possibile utilizzare Index Advisor per determinare le dimensioni di griglia appropriate.

Prerequisiti:

Prima di analizzare i dati che si desidera indicizzare:

- L’ID utente utilizzato dovrà disporre del privilegio SELECT per la tabella.
- Se la tabella contiene più di un milione di righe, utilizzare la clausola ANALYZE per analizzare una serie secondaria di righe in modo che il tempo di elaborazione non sia eccessivo. Per utilizzare la clausola ANALYZE, è necessario disporre del tablespace USER TEMPORARY. La dimensione della pagina del tablespace dovrà essere di almeno 8 KB ed assicurarsi di avere privilegio USE. Ad esempio, le seguenti istruzioni DDL consentono di creare un bufferpool con la stessa dimensione di pagina del tablespace temporaneo dell’utente e garantiscono il privilegio USE a tutti gli utenti:

```
CREATE BUFFERPOOL bp8k SIZE 1000 PAGESIZE 8 K;
CREATE USER TEMPORARY TABLESPACE usertempts
  PAGESIZE 8K
  MANAGED BY SYSTEM USING ('c:\tempts')
  BUFFERPOOL bp8k
GRANT USE OF TABLESPACE usertempts TO PUBLIC;
```

In alternativa, utilizzare il Centro di controllo DB2 per creare un table space e il bufferpool corrispondente.

Procedura:

Per determinare le dimensioni appropriate della griglia di un indice di griglia spaziale:

1. Impostare Index Advisor in modo che suggerisca le dimensioni delle celle della griglia per l’indice che si desidera creare.
 - a. Immettere il comando che richiama Index Advisor con la parola chiave ADVISE per richiedere le dimensioni delle celle della griglia. Ad esempio, per richiamare Index Advisor per la colonna SHAPE della tabella COUNTIES, immettere:

```
gseidx CONNECT TO miodb USER userID USING password GET GEOMETRY
STATISTICS FOR COLUMN userID.counties(shape) ADVISE
```

Limitazione: immettendo il comando **gseidx** da un prompt dei comandi di un sistema operativo, sarà necessario specificare il comando su una sola riga. In alternativa, è possibile eseguire i comandi **gseidx** da un file CLP, che consente di suddividere il comando su più righe.

Index Advisor restituisce le dimensioni delle celle consigliate. Ad esempio, il comando **gseidx** con la parola chiave ADVISE restituisce le seguenti celle consigliate per la colonna shape:

Uso degli indici e delle viste

Dimensione finestra interrogazione	Dimensioni griglia suggerite				Costo
0,1	0,7	2,8	14,0	2,7	
0,2	0,7	2,8	14,0	2,9	
0,5	1,4	3,5	14,0	3,5	
1	1,4	3,5	14,0	4,8	
2	1,4	3,5	14,0	8,2	
5	1,4	3,5	14,0	24	
10	2,8	8,4	21,0	66	
20	4,2	14,7	37,0	190	
50	7,0	14,0	70,0	900	
100	42,0	0,	0	2800	

- b. Scegliere la dimensione appropriata per la finestra di interrogazione dall'output di **gseidx**, in base alla larghezza delle coordinate visualizzate sullo schermo.

In questo esempio, i valori di latitudine e longitudine espressi in gradi decimali, rappresentano le coordinate. Se la visualizzazione della carta geografica generalmente ha una larghezza di circa 0,5 gradi (approssimativamente 55 chilometri), andare alla riga avente valore 0,5 nella colonna Dimensione finestra di interrogazione. Questa riga suggerisce le dimensioni 1,4, 3,5 e 14,0 per la griglia.

2. Creare l'indice utilizzando le dimensioni di griglia suggerite. Per l'esempio riportato precedentemente, è possibile eseguire la seguente istruzioni SQL:

```
CREATE INDEX counties_shape_idx ON userID.counties(shape)
EXTEND USING DB2GSE.SPATIAL_INDEX(1.4,3.5,14.0);
```

Argomenti correlati:

- "Considerazioni sul numero di livelli di indice e dimensioni della griglia" a pagina 102
- "Indici di griglia spaziali" a pagina 100

Attività correlate:

- "Analisi delle statistiche dell'indice di griglia spaziale" a pagina 112

Riferimenti correlati:

- "Istruzione CREATE INDEX per l'indice di griglia spaziale" a pagina 108
- "Comando gseidx" a pagina 117
- "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122

Analisi delle statistiche dell'indice di griglia spaziale

Le statistiche relative a un indice di griglia spaziale esistente possono indicare se l'indice è efficace o se è il caso di sostituirlo con un indice migliore. Per ottenere tali statistiche e, se necessario, sostituire l'indice, utilizzare Index Advisor.

Suggerimento: oltre all'ottimizzazione dell'indice è molto importante anche verificare che venga utilizzato dalle interrogazioni. Per determinare se l'indice è utilizzato o meno, eseguire Visual Explain nel Centro di controllo DB2 oppure utilizzare il comando **db2exfmt** nell'interrogazione. Nella sezione "Access Plan" dell'output esplicativo, se viene visualizzato un operatore EISCAN e il nome dell'indice spaziale, allora l'interrogazione sta utilizzando l'indice.

Prerequisiti:

Prima di analizzare i dati che si desidera indicizzare:

- L'ID utente utilizzato dovrà disporre del privilegio SELECT per la tabella.
- Se la tabella contiene più di un milione di righe, utilizzare la clausola ANALYZE per analizzare una serie secondaria di righe in modo che il tempo di elaborazione non sia eccessivo. Per utilizzare la clausola ANALYZE, è necessario disporre del tablespace USER TEMPORARY. La dimensione della pagina del tablespace dovrà essere di almeno 8 KB ed assicurarsi di avere privilegio USE. Ad esempio, le seguenti istruzioni DDL consentono di creare un bufferpool con la stessa dimensione di pagina del tablespace temporaneo dell'utente e garantiscono il privilegio USE a tutti gli utenti:

```
CREATE BUFFERPOOL bp8k SIZE 1000 PAGESIZE 8 K;
CREATE USER TEMPORARY TABLESPACE usertempts
    PAGESIZE 8K
    MANAGED BY SYSTEM USING ('c:\tempts')
    BUFFERPOOL bp8k
GRANT USE OF TABLESPACE usertempts TO PUBLIC;
```

In alternativa, utilizzare il Centro di controllo DB2 per creare un table space e il bufferpool corrispondente.

Procedura:

Per ottenere le statistiche relative a un indice spaziale e, se necessario, sostituirlo, procedere come segue:

1. Impostare Index Advisor in modo che raccolga le statistiche in base alle dimensioni delle celle dell'indice esistente. E' possibile richiedere le statistiche di tutti i dati indicizzati o solo di una serie secondaria di dati.
 - Per ottenere le statistiche su dati indicizzati in una serie secondaria di righe, immettere il comando **gseidx** e specificare la parola chiave ANALYZE con i relativi parametri, oltre alla clausola indice esistente e alla parola chiave DETAIL. E' possibile specificare il numero o la percentuale di righe che verranno analizzate da Index Advisor per ottenere le statistiche richieste. Ad esempio, per ottenere le statistiche di una serie secondarie di dati indicizzati dall'indice COUNTIES_SHAPE_IDX, immettere:

```
gseidx CONNECT TO miodb USER userID USING password GET GEOMETRY
STATISTICS FOR INDEX userID.counties_shape_idx DETAIL ANALYZE 25 PERCENT
ADVISE
```

- Per ottenere le statistiche di tutti i dati indicizzati, immettere il comando **gseidx** e specificare la relativa clausola di indice esistente. Includere la parola chiave DETAIL. Ad esempio, per richiamare Index Advisor per l'indice COUNTIES_SHAPE_IDX, immettere:

```
gseidx CONNECT TO mydb USER userID USING password GET GEOMETRY
STATISTICS FOR INDEX userID.counties_shape_idx DETAIL SHOW HISTOGRAM ADVISE
```

Index Advisor restituirà le statistiche, un istogramma dei dati e le dimensioni delle celle consigliate per l'indice esistente. Ad esempio, il comando **gseidx** per tutti i dati indicizzati da COUNTIES_SHAPE_IDX restituisce le statistiche seguenti:

Livello griglia 1

```
Dimensione griglia           : 0,5
Numero di forme geometriche  : 2936
Numero di voci di indice     : 12197
```

```
Numero di celle occupate     : 2922
Rapporto voci indice/forme geometriche : 4,154292
```

Uso degli indici e delle viste

Rapporto forma geometrica/celle griglia : 1,004791
Numero massimo di forme geometriche per celle della griglia: 14
Numero minimo di forme geometriche per celle della griglia: 1

Voci di indice:	1	2	3	4	10
Assoluto	: 86	564	72	1519	695
Percentuale (%)	: 2,3	19,21	2,45	51,74	23,67

Livello griglia 2

Dimensione griglia : 0,0
Nessuna forma geometriche indicizzata in questo livello.

Livello griglia 3

Dimensione griglia : 0,0
Nessuna forma geometriche indicizzata in questo livello.

Livello griglia X

Numero di forme geometriche : 205
Numero di voci di indice : 205

2. Stabilire il livello in base al quale le dimensioni delle celle dell'indice esistente faciliteranno il recupero, Valutare le statistiche restituite.

Suggerimenti:

- La statistica "Rapporto voci indice/forme geometriche" dovrebbe essere rappresentata da un valore compreso tra 1 e 4, preferibilmente valori più vicini a 1.
- Il numero di voci di indice per forma geometrica deve essere inferiore a 10 per la dimensione di griglia maggiore, in modo da evitare livelli di overflow. L'aspetto della sezione "Livello griglia X" nell'output di Index Advisor indica che esiste un livello di overflow.

Le statistiche di indice ottenute precedentemente per COUNTIES_SHAPE_IDX indicano che le dimensioni di griglia (0,5, 0, 0) non sono appropriate per i dati di questa colonna, per i seguenti motivi:

- Per il Livello di griglia 1, il valore "Rapporto voci indice/forme geometriche" di 4,154292 è maggiore del valore suggerito, 4.

La riga "Voci di indice" contiene i valori 1, 2, 3, 4 e 10, che indicano il numero di voci di indice per ciascuna forma geometrica. I valori "Assoluti" al di sotto della colonna "Voci di indice", indicano il numero delle forme geometriche che hanno tale numero di voci di indice. Ad esempio, l'output precedente, indica che 1519 forme geometriche hanno 4 voci di indice. Il valore "Assoluto" per 10 voci di indice è 695, il che indica che 695 forme geometriche hanno tra 5 e 10 voci di indice.

- L'aspetto della sezione "Livello griglia X" indica che esiste un livello di indice di overflow. Le statistiche mostrano che 205 forme geometriche hanno più di 10 voci di indice ciascuna.
3. Se le statistiche non sono soddisfacenti, consultare la sezione "Istogramma" e alle righe appropriate nella "Dimensione finestra interrogazione" e alle colonne "Dimensioni griglia suggerite" nell'output di Index Advisor:

- a. Individuare la dimensione dell'MBR avente il numero più elevato di forme geometriche. La sezione "Istogramma" contiene le dimensioni degli MBR e il numero di forme geometriche aventi tale dimensione MBR. Nel seguente istogramma di esempio, il numero più alto di forme geometriche (437) si trova nella dimensione MBR di 0,5.

Istogramma:

Dimensione MBR	Conteggio forme geometriche
0,040000	1
0,045000	3
0,050000	1
0,055000	3
0,060000	3
0,070000	4
0,075000	3
0,080000	4
0,085000	1
0,090000	2
0,095000	1
0,150000	10
0,200000	9
0,250000	15
0,300000	23
0,350000	83
0,400000	156
0,450000	282
0,500000	437
0,550000	397
0,600000	341
0,650000	246
0,700000	201
0,750000	154
0,800000	120
0,850000	66
0,900000	79
0,950000	59
1,000000	47
1,500000	230
2,000000	89
2,500000	34
3,000000	10
3,500000	5
4,000000	3
5,000000	3
5,500000	2
6,000000	2
6,500000	3
7,000000	2
8,000000	1
15,000000	3
25,000000	2
30,000000	1

- b. Passare alla riga della Dimensione finestra di interrogazione avente il valore 0,5 in modo da ottenere le dimensioni consigliate per la griglia (1,4, 3,5, 14,0).

Dimensione finestra interrogazione	Dimensioni griglia suggerite				Costo
0,1	0,7	2,8	14,0	2,7	
0,2	0,7	2,8	14,0	2,9	
0,5	1,4	3,5	14,0	3,5	
1	1,4	3,5	14,0	4,8	
2	1,4	3,5	14,0	8,2	
5	1,4	3,5	14,0	24	
10	2,8	8,4	21,0	66	

Uso degli indici e delle viste

20	4,2	14,7	37,0	190
50	7,0	14,0	70,0	900
100	42,0	0,	0	2800

4. Verificare che le dimensioni consigliate soddisfino le indicazioni riportate al punto 2. Eseguire il comando **gseidx** con le dimensioni consigliate.

```
gseidx CONNECT TO mydb USER userID USING password GET GEOMETRY
STATISTICS FOR COLUMN userID.counties(shape) USING GRID SIZES (1.4, 3.5, 14.0)
```

Livello griglia 1

Dimensione griglia : 1,4
Numero di forme geometriche : 3065
Numero di voci di indice : 5951

Numero di celle occupate : 513
Rapporto voci indice/forme geometriche : 1,941599
Rapporto forma geometrica/celle griglia : 5,974659
Numero massimo di forme geometriche per celle della griglia: 42
Numero minimo di forme geometriche per celle della griglia: 1

Voci di indice:	1	2	3	4	10
-----	-----	-----	-----	-----	-----
Assoluto	: 1180	1377	15	493	0
Percentuale (%)	: 38,50	44,93	0,49	16,08	0,00

Livello griglia 2

Dimensione griglia : 3,5
Numero di forme geometriche : 61
Numero di voci di indice : 143

Numero di celle occupate : 56
Rapporto voci indice/forme geometriche : 2,344262
Rapporto forma geometrica/celle griglia : 1,089286
Numero massimo di forme geometriche per celle della griglia: 10
Numero minimo di forme geometriche per celle della griglia: 1

Voci di indice:	1	2	3	4	10
-----	-----	-----	-----	-----	-----
Assoluto	: 15	28	0	18	0
Percentuale (%)	: 24,59	45,90	0,00	29,51	0,00

Livello griglia 3

Dimensione griglia : 14,0
Numero di forme geometriche : 15
Numero di voci di indice : 28

Numero di celle occupate : 9
Rapporto voci indice/forme geometriche : 1,866667
Rapporto forma geometrica/celle griglia : 1,666667
Numero massimo di forme geometriche per celle della griglia: 10
Numero minimo di forme geometriche per celle della griglia: 1

Voci di indice:	1	2	3	4	10
-----	-----	-----	-----	-----	-----
Assoluto	: 1180	1377	15	493	0
Percentuale (%)	: 46,67	33,33	6,67	13,33	0,00

A questo punto le statistiche mostreranno i valori all'interno delle indicazioni fornite:

- I valori “Rapporto voce di indice/forma geometrica” sono 1,941599 per il Livello di griglia 1, 2,344262 per il Livello di griglia 2 e 1,866667 per il Livello di griglia 3. Questi valori sono compresi tutti nell’intervallo di valori suggeriti da 1 a 4.
 - L’assenza della sezione “Livello di griglia X” indica che il livello di overflow non contiene alcuna voce di indice.
5. Eliminare l’indice esistente e sostituirlo con un indice che indichi le dimensioni di griglia consigliate. Per l’esempio precedente, eseguire le seguenti istruzioni DDL:

```
DROP INDEX userID.counties_shape_idx;
CREATE INDEX counties_shape_idx ON userID.counties(shape) EXTEND USING
DB2GSE.SPATIAL_INDEX(1.4,3.5,14.0);
```

Argomenti correlati:

- “Considerazioni sul numero di livelli di indice e dimensioni della griglia” a pagina 102
- “Indici di griglia spaziali” a pagina 100

Attività correlate:

- “Determinazione delle dimensioni della griglia per un indice di griglia spaziale” a pagina 111
- “Creazione degli indici di griglia spaziali” a pagina 106

Riferimenti correlati:

- “Istruzione CREATE INDEX per l’indice di griglia spaziale” a pagina 108
- “Comando gseidx” a pagina 117
- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

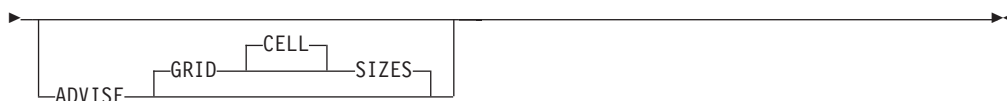
Comando gseidx

Utilizzare il comando **gseidx** per richiamare Index Advisor per gli indici di griglia spaziali.

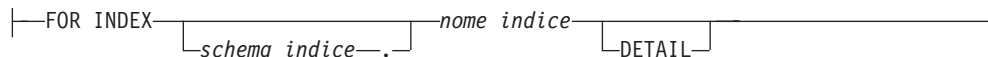
Sintassi

```
► gseidx CONNECT TO nome_database [USER id_utente USING password]
► GET GEOMETRY STATISTICS [indice esistente | indice simulato]
► [ANALYZE numero ROWS ONLY]
  [PERCENT (1)]
► [SHOW [MINIMUM BOUNDING RECTANGLE] HISTOGRAM [WITH n BUCKETS]]
```

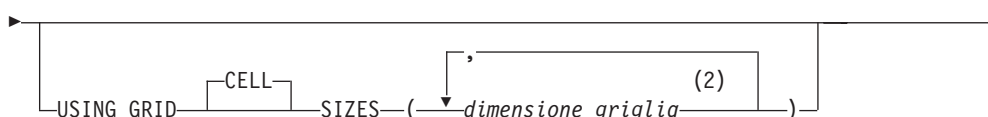
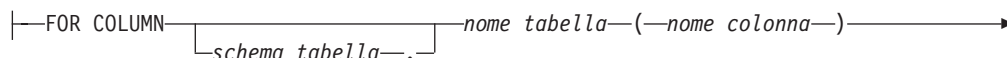
Uso degli indici e delle viste



indice esistente:



indice simulato:



Note:

- 1 Invece della parola PERCENT, è possibile specificare un simbolo di percentuale (%).
- 2 E' possibile specificare le dimensioni delle celle per uno, due o tre livelli di griglia.

Parametri:

nome_database

Il nome del database che contiene la tabella spaziale.

id utente

L'ID utente con autorizzazione SYSADM o DBADM per il database in cui si trova l'indice o la tabella o autorizzazione SELECT per la tabella. Se si accede all'ambiente di comandi DB2 con l'ID utente del proprietario del database, non sarà necessario specificare l'*id utente* e la *password* nel comando **gseidx**.

password

Password per l'ID utente.

indice esistente:

L'indice esistente nel quale raccogliere le statistiche.

schema indice

Il nome dello schema che contiene l'indice esistente.

nome indice

Nome dell'indice esistente.

DETAIL

Mostra le informazioni su ciascun livello di griglia:

- La dimensione delle celle della griglia
- Il numero di forme geometriche indicizzate
- Il numero di voci dell'indice
- Il numero di celle della griglia che contengono le forme geometriche

- Il numero medio di voci della griglia per forma geometrica
- Il numero medio di forme geometriche per ciascuna cella della griglia
- Il numero di forme geometriche nella cella che contiene più forme
- Il numero di forme geometriche nella cella che contiene meno forme

indice simulato:

Si riferisce ad una colonna della tabella e al relativo indice simulato.

schema tabella

Nome dello schema che contiene la tabella con la colonna alla quale si riferisce l'indice simulato.

nome tabella

Nome della tabella che contiene la colonna alla quale si riferisce l'indice simulato.

nome colonna

Nome della colonna alla quale si riferisce l'indice simulato.

dimensione griglia

Le dimensioni delle celle in ciascun livello della griglia (livello massimo, medio e minimo) di un indice simulato. E' necessario specificare la dimensione delle celle per almeno un livello. Se non si desidera includere alcun livello, non specificare alcuna dimensione o indicare dimensione (0,0).

Indicando il parametro *dimensione griglia*, Index Advisor restituirà lo stesso tipo di statistiche restituite includendo la parola chiave **DETAIL** nella clausola dell'indice esistente.

ANALYZE numero ROWS | PERCENT ONLY

Ottiene le statistiche relative ai dati indicizzati in una serie secondaria di righe della tabella. Se la tabella contiene più di un milione di righe, utilizzare la clausola **ANALYZE** in modo che il tempo di elaborazione non sia eccessivo. Indicare la quantità o la percentuale approssimativa delle righe da includere nella serie secondaria.

SHOW MINIMUM BOUNDING RECTANGLE HISTOGRAM

Consente di visualizzare un diagramma contenente gli MBR (minimum bounding rectangle) delle forme geometriche e il numero di forme i cui MBR hanno la stessa dimensione.

WITH n BUCKETS

Indica il numero di raggruppamenti degli MBR di tutte le forme geometriche analizzate. Gli MBR più piccoli vengono raggruppati con le forme geometriche di dimensioni ridotte. Gli MBR più grandi vengono raggruppati con le altre forme geometriche di dimensioni maggiori.

Se questo parametro non viene specificato o si indicano 0 settori, Index Advisor visualizzerà le dimensioni dei settori in formato logaritmico. Ad esempio, le dimensioni dell'MBR possono essere valori logaritmici, quali 1,0, 2,0, 3,0,... 10,0, 20,0, 30,0,... 100,0, 200,0, 300,0,...

Se si specifica un numero di settori maggiore di 0, Index Advisor visualizzerà valori di uguale dimensione. Ad esempio, le dimensioni dell'MBR potrebbero essere valori di uguale dimensione, quali 8,0, 16,0, 24,0,... 320,0, 328,0, 334,0.

Per impostazione predefinita, vengono utilizzati settori di dimensione logaritmica.

Uso degli indici e delle viste

ADVISE GRID CELL SIZES

Elabora le dimensioni migliori per le celle della griglia.

Note di utilizzo:

Se il comando **gseidx** viene eseguito da un prompt di comandi di un sistema operativo, sarà necessario l'intero comando su una singola riga.

Esempio:

Nell'esempio seguente viene mostrata una richiesta di informazioni su un indice di griglia esistente il cui nome completo è COUNTIES_SHAPE_IDX e vengono suggerite le dimensioni per l'indice di griglia appropriate:

```
gseidx CONNECT TO miodb USER ID utente USING password GET GEOMETRY  
STATISTICS FOR INDEX IDutente.counties_shape_idx DETAIL ADVISE
```

Per una spiegazione delle informazioni restituite da questo comando, consultare la sezione "Analisi delle statistiche dell'indice di griglia spaziale" a pagina 112.

Argomenti correlati:

- "Considerazioni sul numero di livelli di indice e dimensioni della griglia" a pagina 102
- "Indici di griglia spaziali" a pagina 100
- "Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica" a pagina 110

Attività correlate:

- "Determinazione delle dimensioni della griglia per un indice di griglia spaziale" a pagina 111
- "Creazione degli indici di griglia spaziali" a pagina 106

Riferimenti correlati:

- "Istruzione CREATE INDEX per l'indice di griglia spaziale" a pagina 108
- "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122

Utilizzo delle viste per accedere alle colonne spaziali

E' possibile definire una vista che utilizzi una colonna spaziale allo stesso modo in cui vengono definite le viste in DB2 per altri tipi di dati. Se si dispone di una tabella con una colonna spaziale e si desidera una vista per utilizzare la colonna, attenersi alle seguenti informazioni.

Attività correlate:

- "Creating a view" nella *Administration Guide: Implementation*

Riferimenti correlati:

- "CREATE VIEW statement" nella *SQL Reference, Volume 2*

Capitolo 12. Analisi e generazione delle informazioni spaziali

Dopo aver inserito i dati nelle colonne spaziali, è possibile eseguire l'interrogazione. Il presente capitolo:

- Descrive gli ambienti in cui è possibile inoltrare le interrogazioni
- Fornisce esempi sui vari tipi di funzioni spaziali che è possibile richiamare in una interrogazione
- Fornisce istruzioni per l'uso delle funzioni spaziali con gli indici spaziali

Ambienti di esecuzione delle analisi spaziali

E' possibile eseguire l'analisi dei dati utilizzando le funzioni SQL e spaziali nei seguenti ambienti di programmazione:

- Istruzioni SQL interattive.
È possibile immettere istruzioni SQL interattive dall'Editor comandi DB2®, dalla finestra Comandi DB2 oppure dal processore di riga comandi DB2.
- Programmi applicativi in tutte le lingue supportate da DB2.

Esempi delle operazioni eseguite dalle funzioni spaziali

DB2 Spatial Extender fornisce funzioni in grado di eseguire diverse operazioni con i dati spaziali. Tali funzioni possono essere suddivise in base al tipo di operazione che eseguono. La Tabella 5 contiene le categorie in cui sono suddivise le funzioni e alcuni esempi. Il testo riportato dopo la Tabella 5 mostra il codice utilizzato per gli esempi.

Tabella 5. Operazioni e funzioni spaziali

Categoria della funzione	Operazione di esempio
Restituisce le informazioni su forme geometriche specifiche.	Restituire l'area, in miglia quadrate, dell'area Vendite del Negozio 10.
Effettua confronti.	Determinare se la posizione dell'abitazione di un cliente è prossima all'area Vendite del Negozio 10.
Ricava nuove forme geometriche da forme esistenti.	Ricavare l'area Vendite di un negozio dalla rispettiva posizione.
Converte le forme geometriche in e da formati di scambio dati.	Convertire le informazioni di un cliente in formato GML in una forma geometrica, in modo che le informazioni possano essere aggiunte ad un database DB2.

Esempio 1: Restituisce le informazioni su forme geometriche specifiche:

In questo esempio, la funzione ST_Area restituisce un valore numerico che rappresenta l'area Vendite del Negozio 10. L'area sarà restituita nella stessa unità di misura del sistema di coordinate utilizzato per definire la posizione dell'area.

```
SELECT db2gse.ST_Area(sales_area)
FROM stores
WHERE id = 10
```

Generazione e analisi delle informazioni spaziali

Il seguente esempio mostra l'operazione precedente, ma questa volta la funzione ST_Area viene richiamata come metodo e restituirà l'area in miglia quadrate.

```
SELECT sales_area..ST_Area('STATUTE MILE')
FROM   stores
WHERE  id = 10
```

Esempio 2: Effettua confronti:

In questo esempio, la funzione ST_Within paragonerà le coordinate della forma geometrica che rappresenta l'abitazione di un cliente con le coordinate di una forma geometrica che rappresenta l'area Vendite del Negozio 10. Il risultato della funzione indicherà se l'abitazione è prossima all'area vendite.

```
SELECT c.first_name, c.last_name, db2gse.ST_Within(c.location, s.sales_area)
FROM   customers as c. stores AS s
WHERE  s.id = 10
```

Esempio 3: Ricava nuove forme geometriche da forme esistenti:

In questo esempio, la funzione ST_Buffer ricava una forma geometrica che rappresenta l'area Vendite di un negozio da una forma geometrica che rappresenta la posizione del negozio.

```
UPDATE stores
SET    sales_area = db2gse.ST_Buffer(location, 10, 'KILOMETERS')
WHERE  id = 10
```

Il seguente esempio mostra l'operazione precedente, ma questa volta la funzione ST_Buffer viene richiamata come metodo.

```
UPDATE stores
SET    sales_area = location..ST_Buffer(10, 'KILOMETERS')
WHERE  id = 10
```

Esempio 4: Converti le forme geometriche in e da formati di scambio dati:

In questo esempio, le informazioni sul cliente codificate in GML vengono convertite in una forma geometrica, in modo che possano essere memorizzate in un database DB2.

```
INSERT
INTO   c.name,c.phoneNo,c.address
VALUES ( 123, 'Mary Anne', Smith', db2gse.ST_Point('
<gml:Point><gml:coord><gml:X>-130.876</gml:X>
<gml:Y>41.120'</gml:Y></gml:coord></gml:Point>, 1) )
```

Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni

Il gruppo di funzioni spaziali, definito *funzioni di confronto*, è in grado di migliorare le prestazioni delle interrogazioni sfruttando gli indici di griglia spaziali o gli indici geodetici Voronoi (noti entrambi come *indici spaziali*). Ciascuna di queste funzioni confronta due forme geometriche. Se il risultato del confronto soddisfa determinati criteri, la funzione restituisce valore 1; in caso contrario restituisce valore 0. Se non è possibile eseguire il confronto, la funzione restituisce un valore nullo.

Ad esempio, la funzione ST_Overlaps confronta due forme geometriche con la stessa dimensione (ad esempio, due linee o due poligoni). Se le forme si sovrappongono solo in parte e l'area dell'intersezione ha la stessa dimensione delle due forme, ST_Overlaps restituisce valore 1.

La Tabella 6 mostra le funzioni di confronto che possono utilizzare l'indice di griglia spaziale e quelle che possono utilizzare l'indice geodetico Voronoi:

Tabella 6. Funzioni di confronto che possono utilizzare l'indice di griglia spaziale o l'indice geodetico Voronoi

Funzioni di confronto	Possono utilizzare l'indice di griglia spaziale	Possono utilizzare l'indice geodetico Voronoi
EnvelopesIntersect	Sì	Sì
ST_Contains	Sì	Sì
ST_Crosses	Sì	No
ST_Distance	Sì	Sì
ST_EnvIntersects	Sì	Sì
ST_Equals	Sì	No
ST_Intersects	Sì	Sì
ST_MBRIntersects	Sì	Sì
ST_Overlaps	Sì	No
ST_Touches	Sì	No
ST_Within	Sì	Sì

A causa della quantità di memoria e del tempo necessari per eseguire una funzione, tale esecuzione potrebbe richiedere un'elaborazione considerevole. Inoltre, più complesse sono le geometrie da confrontare, più difficile e lungo sarà il processo di confronto. Le funzioni specializzate indicate in precedenza, sono in grado di completare le operazioni più velocemente utilizzando un indice spaziale per individuare le forme geometriche. Per abilitare le funzioni all'utilizzo dell'indice spaziale, attenersi alle seguenti regole:

- Specificare la funzione in una clausola WHERE. Se viene specificata nelle clausole SELECT, HAVING o GROUP BY, non sarà possibile utilizzare l'indice spaziale.
- La funzione dovrà essere l'espressione a sinistra del predicato.
- L'operatore utilizzato nel predicato che confronta il risultato della funzione con un'altra espressione deve essere il segno uguale con una eccezione: la funzione ST_Distance dovrà utilizzare il segno meno.
- L'espressione a destra del predicato deve essere la costante 1, tranne quando ST_Distance è la funzione a sinistra.
- L'operazione deve coinvolgere una ricerca nella colonna spaziale in cui è stato definito l'indice spaziale.

Ad esempio:

```
SELECT c.name, c.address, c.phone
FROM customers AS c, bank_branches AS b
WHERE db2gse.ST_Distance(c.location, b.location) < 10000
and b.branch_id = 3
```

La Tabella 7 a pagina 124 indica i metodi di creazione delle interrogazioni spaziali corretti e quelli non validi per utilizzare un indice spaziale.

Generazione e analisi delle informazioni spaziali

Tabella 7. Come le funzioni spaziali possono rispettare o violare le regole per l'utilizzo di un indice spaziale.

Interrogazioni che si riferiscono alle funzioni spaziali	Regole non rispettate
<pre>SELECT * FROM stores AS s WHERE db2gse.ST_Contains(s.sales_zone, ST_Point(-121.8,37.3, 1)) = 1</pre>	Nessuna condizione è stata violata in questo esempio.
<pre>SELECT * FROM stores AS s WHERE db2gse.ST_Length(s.location) > 10</pre>	La funzione spaziale ST_Length non esegue un confronto di forme geometriche, quindi non può utilizzare un indice spaziale.
<pre>SELECT * FROM stores AS s WHERE 1=db2gse.ST_Within(s.location,:BayArea)</pre>	La funzione deve essere un'espressione a sinistra del predicato.
<pre>SELECT * FROM stores AS s WHERE db2gse.ST_Contains(s.sales_zone, ST_Point(-121.8,37.3, 1)) <> 0</pre>	I confronti di uguaglianza devono utilizzare la costante 1.
<pre>SELECT * FROM stores AS s WHERE db2gse.ST_Contains(ST_Polygon ('polygon((10 10, 10 20, 20 20, 20 10, 10 10))', 1), ST_Point(-121.8, 37.3, 1)) = 1</pre>	Non esiste alcun indice spaziale in nessuno degli argomenti della funzione, quindi non è possibile utilizzare un indice.

Argomenti correlati:

- “Considerazioni sul numero di livelli di indice e dimensioni della griglia” a pagina 102
- “Indici geodetici Voronoi” a pagina 175
- “Indici di griglia spaziali” a pagina 100
- “Regolazione degli indici di griglia spaziali con Index Advisor—Panoramica” a pagina 110

Attività correlate:

- “Creazione di indici geodetici Voronoi” a pagina 179
- “Creazione degli indici di griglia spaziali” a pagina 106

Capitolo 13. Comandi di DB2 Spatial Extender

Questo capitolo descrive i comandi utilizzati per impostare DB2 Spatial Extender. Inoltre spiega come utilizzare tali comandi durante lo sviluppo dei progetti.

Richiamo dei comandi per l'impostazione di DB2 Spatial Extender e per lo sviluppo di progetti

Per impostare Spatial Extender e creare progetti basati su dati spaziali, è possibile utilizzare il processore riga comandi db2se. Questa sezione descrive come utilizzare db2se per eseguire i comandi di DB2 Spatial Extender.

Prerequisiti:

Per poter utilizzare i comandi db2se, è necessario disporre delle autorizzazioni appropriate. Per individuare il tipo di autorizzazione richiesta per un determinato comando, fare riferimento alla Tabella 8 e all'argomento della procedura memorizzata per il comando. Ad esempio, il comando **db2se create_srs** richiede le stesse autorizzazioni della procedura memorizzata db2.ST_create_srs.

Eccezione: Il comando **db2se shape_info** non richiama alcuna procedura memorizzata, ma consente di visualizzare informazioni sul contenuto dei file shape.

Procedura:

Immettere i comandi db2se da un prompt dei comandi del sistema operativo.

Per ottenere maggiori informazioni sui comandi secondari e sui parametri che è possibile specificare, procedere come segue:

- Immettere db2se o db2se -h e premere Invio. Viene visualizzato un elenco di comandi secondari di db2se.
- Immettere db2se e un comando secondario, oppure db2se e un comando secondario seguito da -h. Premere Invio. La sintassi fa sì che il comando secondario venga visualizzato. In questa sintassi:
 - Ciascun parametro è preceduto da un trattino e seguito da un indicatore di posizione per il valore di un parametro.
 - I parametri racchiusi tra parentesi sono facoltativi. Gli altri parametri sono obbligatori.

Importante: La sintassi dei comandi può essere richiamata interattivamente sul proprio monitor, in modo che non sia necessario ricercare altrove tali informazioni.

Per utilizzare un comando db2se, immettere db2se. Quindi immettere il comando secondario, seguito dai parametri e dai valori dei parametri richiesti. Premere Invio.

Nelle versioni precedenti, il prefisso dei comandi secondari era gseadm invece di db2se. Gli script gseadm creati nelle versioni precedenti continueranno a

funzionare nella versione 8.1, ma IBM suggerisce di migrare gli script in modo da poter utilizzare il processore di riga comandi db2se.

Tenere presente che:

- Per accedere al database specificato potrebbe essere necessario indicare ID utente e password. Ad esempio, se si desidera accedere al database come un altro utente, indicare i nuovi ID utente e password. L'ID utente e la password devono essere sempre preceduti, rispettivamente, dai parametri userId e pw.

Se non viene indicato alcun ID utente o password, per impostazione predefinita verranno utilizzati l'ID utente e password dell'utente corrente.

- Per impostazione predefinita, i valori immessi non rilevano la distinzione tra maiuscolo e minuscolo. Per far sì che i valori rilevino la distinzione tra maiuscolo e minuscolo, racchiuderli tra doppi apici. Ad esempio, per indicare il nome in caratteri minuscoli della tabella miatabella, immettere: "miatabella".

Nota: per assicurarsi che i doppi apici non vengano interpretati dal prompt del sistema (shell) specificare, ad esempio, quanto segue:

```
\ "miatabella\"
```

Se un valore sensibile al maiuscolo e minuscolo è qualificato da un altro valore ugualmente sensibile, delimitare i due valori individualmente; ad esempio:

```
"mioschema"."miatabella"
```

Racchiudere le stringhe in doppi apici, ad esempio:

```
"select * from newtable"
```

Una volta eseguito il comando db2se, viene richiamata la procedura memorizzata che corrisponde al comando, che eseguirà l'operazione richiesta.

Panoramica dei comandi di db2se:

La seguente tabella indica i comandi db2se da utilizzare per eseguire l'impostazione di Spatial Extender e creare progetti che utilizzano dati spaziali. La tabella contiene, inoltre, alcuni esempi dei comandi db2se e riferimenti alle autorizzazioni e ai parametri relativi a ciascun comando. Nella parte destra della tabella, nella seconda colonna, viene riportato un collegamento o un riferimento alle informazioni relative alla procedura memorizzata richiamata nel momento in cui viene utilizzato il comando. L'autorizzazione necessaria per utilizzare la procedura memorizzata sarà la stessa per l'utilizzo del comando; inoltre, il comando e la procedura condividono gli stessi parametri. Per ulteriori informazioni sull'autorizzazione e il significato dei diversi parametri, consultare la sezione indicata dal riferimento.

Tabella 8. Comandi db2se ordinati in base all'attività

Attività	Comando ed esempio
Creazione di un sistema di coordinate	<p>db2se create_cs</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_create_coordsys.</p> <p>Nell'esempio seguente viene creato un sistema di coordinate chiamato "miosistemacoord".</p> <pre>db2se create_cs miodb -coordsysName \"miosistemacoord\" -definition GEOCS[\"GCS_NORTH_AMERICAN_1983\"], DATUM[\"D_North_American_1983\"], SPHEROID[\"GRS_1980\",6387137,298.257222101]], PRIMEM[\"Greenwich\",0],UNIT[\"Degree\", 0.0174532925199432955]]</pre>
Creazione di un sistema di riferimento spaziale	<p>db2se create_srs</p> <p>I parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_create_srs. Non è richiesta alcuna autorizzazione.</p> <p>Nell'esempio seguente viene creato un sistema di riferimento spaziale chiamato "miosist".</p> <pre>db2se create_srs miodb -srsName \"miosist\" -srsID 100 -xScale 10 -coordsysName \"GCS_North_American_1983\"</pre>
Eliminazione di un sistema di riferimento spaziale	<p>db2se drop_srs</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_drop_srsdb2gse.ST_drop_srs.</p> <p>Nell'esempio seguente viene cancellato un sistema di riferimento spaziale chiamato "miosist".</p> <pre>db2se drop_srs miodb -srsName \"miosist\"</pre>
Eliminazione della definizione di un sistema di coordinate	<p>db2se drop_cs</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_drop_coordsysdb2gse.ST_drop_coordsys.</p> <p>Nell'esempio seguente viene cancellato un sistema di coordinate chiamato "miosistemacoord".</p> <pre>db2se drop_cs miodb -coordsysName \"miosistemacoord\"</pre>
Annullamento della funzione di geocoding automatica	<p>db2se disable_autogc</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_disable_autogeocoding.</p> <p>Nel seguente esempio viene disabilitata la funzione di geocoding automatica per la colonna MIACOLONNA nella tabella MIATABELLA.</p> <pre>db2se disable_autogc miodb -tableName \"miatabella\" -columnName \"miacolonna\"</pre>

Tabella 8. Comandi db2se ordinati in base all'attività (Continua)

Attività	Comando ed esempio
<p>Abilitazione del database per le operazioni spaziali</p>	<p>db2se enable_db</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_enable_dbdb2gse.ST_enable_db.</p> <p>Nel seguente esempio il database MIODB viene abilitato per le operazioni spaziali.</p> <pre>db2se enable_db miodb</pre>
<p>Esportazione dei dati in un file di trasferimento SDE</p>	<p>db2se export_sde</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.GSE_export_sdedb2gse.GSE_export_sde.</p> <p>Nel seguente esempio i dati vengono esportati dalla tabella MIATABELLASDE, che contiene la colonna spaziale MIACOLONNASPAZIALE, nel file di trasferimento SDE miofilesde.</p> <pre>db2se export_sde miodb -tableName \miatabellaSDE\ -columnName \miaColonnaspaziale\ -fileName /home/mioaccount/miofilesde</pre> <p>Nell'esempio successivo, i dati vengono esportati dalla tabella TABELLASPAZIALE nel file SDE sdex, che verrà creato sul client DB2. Gli errori e i messaggi informativi (ad esempio, l'ora di avvio e di completamento dell'esportazione e quante righe sono state esportate) vengono registrati nel file sdex.export.log.</p> <pre>db2se export_sde miodb -client -fileName sdex -selectStatement "SELECT * FROM tabellaSpaziale" -messagesFile sdex.export.log</pre>
<p>Esportazione dei dati in file shape</p>	<p>db2se export_shape</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_export_shape.</p> <p>Nel seguente esempio viene esportata la colonna spaziale MIA COLONNA e la relativa tabella associata MIATABELLA, nel file shape miofilesshape.</p> <pre>db2se export_shape miodb -fileName /home/mioaccount/miofilesshape -selectStatement "select * from miatabella"</pre>

Tabella 8. Comandi db2se ordinati in base all'attività (Continua)

Attività	Comando ed esempio
Importazione di un file di trasferimento SDE	<p>db2se import_sde</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.GSE_import_sdedb2gse.GSE_import_sde.</p> <p>Nel seguente esempio viene importato il file di trasferimento SDE miofilesde nella tabella MIATABELLASDE, che contiene la colonna spaziale MIACOLONNASPAZIALE. Per ogni dieci record elaborati viene eseguito un commit.</p> <pre>db2se import_sde miodb -tableName \"miatabellasde\" -columnName \"miacolonnaspaziale\" -fileName /home/mioaccount/\"miofilesde\" -commitScope 10</pre> <p>L'esempio successivo mostra come importare il file SDE sdex, ubicato sul client DB2. In questo esempio, i dati vengono importati nella tabella TABELLASDE (nella colonna ID) e viene eseguito un commit per ogni 100 record elaborati. Eventuali errori verranno registrati nel file sdex.exceptions.</p> <pre>db2se import_sde miodb -client -filename sdex -srsId 1234 -tableName tabellaSde -idColumn id -commitScope 100 -messagesFile sdex.exceptions</pre>
Importazione di file shape	<p>db2se import_shape</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_import_shape.</p> <p>Il seguente comando consente di importare il file shape miofile nella tabella MIATABELLA. Durante l'importazione, i dati spaziali contenuti nel miofile verranno inseriti nella colonna MIACOLONNA della MIATABELLA.</p> <pre>db2se import_shape mydb -fileName \"miofile\" -srsName NAD83_SRS_1 -tableName \"miatabella\" -spatialColumnName \"miacolonna\"</pre>
Registrazione di un geocoder	<p>db2se register_gc</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_register_geocoder.</p> <p>Nel seguente esempio viene registrato il geocoder "miogeocoder", implementato dalla funzione "mioschema.miafunzione".</p> <pre>db2se register_gc miodb -geocoderName \"miogeocoder\" -functionSchema \"mioschema\" -functionName \"miafunzione\" -defaultParameterValues \"1, 'string',,cast(null as varchar(50))\" -fornitore miofornitore -descrizione \"geocoder del mio fornitore contenente testo noto\"</pre>

Tabella 8. Comandi db2se ordinati in base all'attività (Continua)

Attività	Comando ed esempio
Registrazione di una colonna spaziale	<p>db2se register_spatial_column</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_register_spatial_column.</p> <p>Nel seguente esempio viene registrata la colonna spaziale MIACOLONNA nella tabella MIATABELLA, con il sistema di riferimento spaziale "USA_SRS_1".</p> <pre>db2se register_spatial_column miodb -tableName \"miatabella\" -columnName \"miacolonna\" -srsName USA_SRS_1</pre>
Eliminazione delle risorse che abilitano il database per le operazioni spaziali.	<p>db2se disable_db</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_disable_dbdb2gse.ST_disable_db.</p> <p>Nel seguente esempio le risorse che abilitano il database MIODB per le operazioni spaziali vengono eliminate.</p> <pre>db2se disable_db miodb</pre>
Eliminazione delle impostazioni per le funzioni di geocoding	<p>db2se remove_gc_setup</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_remove_gc_setup.</p> <p>Nel seguente esempio vengono annullate le impostazioni per le funzioni di geocoding relative alla colonna spaziale MIACOLONNA nella tabella MIATABELLA.</p> <pre>db2se remove_geocoding_setup miodb -tableName \"miatabella\" -columnName \"miacolonna\"</pre>
Esecuzione del geocoder in modalità batch	<p>db2se run_gc</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_run_gc.</p> <p>Nel seguente esempio il geocoder viene eseguito in modalità batch per inserire i dati nella colonna MIACOLONNA della tabella MIATABELLA.</p> <pre>db2se run_gc miodb -tableName \"miatabella\" -columnName \"miacolonna\"</pre>
Impostazione del geocoder per l'esecuzione automatica	<p>db2se enable_autogeocoding</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_enable_autogeocoding.</p> <p>Nel seguente esempio viene impostata la funzione di geocoding automatica per la colonna MIACOLONNA nella tabella MIATABELLA:</p> <pre>db2se enable_autogeocoding miodb -tableName \"miatabella\" -columnName \"miacolonna\"</pre>

Tabella 8. Comandi db2se ordinati in base all'attività (Continua)

Attività	Comando ed esempio
Impostazione delle funzioni di geocoding	<p>db2se setup_gc</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_setup_geocoding.</p> <p>Nel seguente esempio vengono impostate le funzioni di geocoding automatico per inserire i dati nella colonna MIACOLONNA nella tabella MIATABELLA:</p> <pre>db2se setup_gc miodb -tableName \"miatabella\" -columnName \"miacolonna\" -geocoderName \"db2se_USA_GEOCODER\" -parameterValues \"address,city,state,zip,2,90,70,20,1.1,'meter',4..\" -autogeocodingColumns address,city,state,zip commitScope 10</pre>
Visualizzazione informazioni su un file shape e il relativo contenuto	<p>db2se shape_info</p> <p>Per utilizzare questo comando, è necessario:</p> <ul style="list-style-type: none"> • Disporre dell'autorizzazione per la lettura del file a cui il comando si riferisce. • Connettersi al database che contiene il file (se si utilizza il parametro <i>-database</i>, che specifica che il sistema deve ricercare nel database indicato i sistemi di coordinate compatibili e i sistemi di riferimento spaziali). <p>Il seguente esempio mostra le informazioni relative al file shape miofile, ubicato nella directory corrente.</p> <pre>db2se shape_info -fileName miofile</pre> <p>Il seguente esempio mostra le informazioni relative a al file shape UNIX di esempio offices. Il parametro <i>-database</i> consente di individuare tutti i sistemi spaziali e di coordinate compatibili nel database specificato (in questo caso, MIODB).</p> <pre>db2se shape_info -fileName ~/sqllib/samples/spatial/data/offices -database mioDB</pre>
Visualizzazione informazioni su un file SDE e il relativo contenuto.	<p>db2se sde_info</p> <p>Per utilizzare questo comando, è necessario:</p> <ul style="list-style-type: none"> • Disporre dell'autorizzazione per la lettura del file a cui il comando si riferisce. • Connettersi al database che contiene il file (se si utilizza il parametro <i>-database</i>, che specifica che il sistema deve ricercare nel database indicato i sistemi di coordinate compatibili e i sistemi di riferimento spaziali). <p>Il seguente esempio mostra le informazioni relative al file SDE filesde, ubicato nella directory corrente.</p> <pre>db2se sde_info -fileName miofile</pre> <p>L'esempio successivo mostra come ottenere informazioni sul file SDE sdex e consente di ricercare nel database MIODB tutti sistemi di riferimento spaziali e di coordinate compatibili.</p> <pre>db2se sde_info -fileName data/sdex -database mioDB</pre>

Tabella 8. Comandi db2se ordinati in base all'attività (Continua)

Attività	Comando ed esempio
Annullamento della registrazione del geocoder	<p>db2se unregister_gc</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_unregister_geocoder.</p> <p>Nel seguente esempio viene annullata la registrazione del geocoder "miogeocoder".</p> <pre>db2se unregister_gc mydb -geocoderName \"mygeocoder\"</pre>
Annullamento della registrazione di una colonna spaziale	<p>db2se unregister_spatial_column</p> <p>Le autorizzazioni e i parametri del comando sono gli stessi necessari per la procedura memorizzata db2gse.ST_unregister_spatial_column.</p> <p>Nel seguente esempio viene annullata la registrazione della colonna spaziale MIACOLONNA nella tabella MIATABELLA.</p> <pre>db2se unregister_spatial_column miodb -tableName \"miatabella\" -columnName \"miacolonna\"</pre>
Aggiornamento della definizione di un sistema di coordinate	<p>db2se alter_cs</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_alter_coordsysdb2gse.ST_alter_coordsys.</p> <p>Nell'esempio seguente viene aggiornata la definizione del sistema di coordinate "miosistemacoord" con un nuovo nome di organizzazione.</p> <pre>db2se alter_cs mydb -coordsysName \"miosistemacoord\" -organization mianuovaorg -tableName \"miatabella\"</pre>
Aggiornamento della definizione di un sistema di riferimento spaziale	<p>db2se alter_srs</p> <p>Le autorizzazioni e i parametri del comandi sono gli stessi necessari per la procedura memorizzata db2gse.ST_alter_srsdb2gse.ST_alter_srs.</p> <p>Nell'esempio seguente viene modificato il sistema di riferimento spaziale "miosist" con un valore xOffset e una descrizione diversi.</p> <pre>db2se alter_srs miodb -srsName \"miosist\" -xoffset 35 -descrizione \"Mio sistema di riferimento spaziale.\"</pre>

Capitolo 14. Scrittura di applicazioni ed utilizzo del programma di esempio

Questo capitolo indica come scrivere le applicazioni di Spatial Extender.

Scrittura di applicazioni per DB2 Spatial Extender

Le seguenti attività di riferimento contengono informazioni utili per l'elaborazione di programmi applicativi che richiamino le procedure memorizzate o le funzioni di DB2 Spatial Extender.

Argomenti correlati:

- "Programma di esempio DB2 Spatial Extender" a pagina 135

Attività correlate:

- "Richiamo delle procedure memorizzate di DB2 Spatial Extender da un'applicazione" a pagina 134
- "Inserimento del file di intestazione di DB2 Spatial Extender nelle applicazioni spaziali" a pagina 133

Inserimento del file di intestazione di DB2 Spatial Extender nelle applicazioni spaziali

DB2 Spatial Extender contiene un file di intestazione che definisce le costanti che è possibile utilizzare con le procedure memorizzate e le funzioni di DB2 Spatial Extender.

Suggerimento: Se si desidera richiamare le procedure memorizzate o le funzioni di DB2 Spatial Extender da programmi C o C++, inserire il file di intestazione nelle applicazioni spaziali.

Procedura:

Per far sì che le applicazioni di DB2 Spatial Extender possano utilizzare le definizioni necessarie contenute in questo file di intestazione, procedere come segue:

1. Inserire il file di intestazione di DB2 Spatial Extender nel programma applicativo utilizzato. Il nome del file di intestazione è:
`db2gse.h`
Il file di intestazione si trova nella directory *percorsodb2/include*, dove *percorsodb2* è la directory di installazione in cui si trova DB2 Universal Database.
2. Verificare che il percorso della directory sia stato indicato nel `makefile` insieme alle opzioni di compilazione.

Se si desidera generare applicazioni Windows a 64-bit su un sistema Windows a 32-bit, modificare il parametro `DB2_LIBS` nel file `samples/spatial/makefile.nt` in modo che si adatti alle applicazioni a 64-bit. Di seguito vengono riportate le modifiche necessarie:

```
DB2_LIBS = $(DB2_DIR)\lib\Win64\db2cli.lib $(DB2_DIR)\lib\Win64\db2api.lib
```

Richiamo delle procedure memorizzate di DB2 Spatial Extender da un'applicazione

Le procedure memorizzate di DB2 Spatial Extender vengono create al momento dell'abilitazione del database per le operazioni spaziali. Se si desidera scrivere programmi applicativi che richiamano le procedure memorizzate di DB2 Spatial Extender, utilizzare l'istruzione SQL CALL e indicare il nome della procedura desiderata.

Procedura:

Per richiamare le procedure memorizzate di DB2 Spatial Extender, procedere come segue:

- Per richiamare la procedura memorizzata ST_enable_db, che abilita il database per le operazioni spaziali, indicare il nome della procedura come segue:

```
CALL db2gse!ST_enable_db
```

Il parametro db2gse! in questo comando, rappresenta il nome della libreria di DB2 Spatial Extender. La procedura memorizzata ST_enable_db è l'unica in cui è necessario inserire un punto esclamativo nel comando (db2gse!).

- Per richiamare le altre procedure di DB2 Spatial Extender, indicare il nome della procedura memorizzata nel seguente formato, dove db2gse è il nome dello schema di tutte le procedure memorizzate di DB2 Spatial Extender e *nome_procedura_spaziale* è il nome della procedura:

```
CALL db2gse.nome_procedura_spaziale
```

Questa volta, il comando non contiene alcun punto esclamativo.

La seguente tabella contiene le procedure memorizzate di DB2 Spatial Extender.

Tabella 9.

Procedura memorizzata	Descrizione
GSE_export_sde	Esporta una colonna spaziale e le tabelle ad essa associate in un file di trasferimento SDE.
GSE_import_sde	Importa un file di trasferimento SDE in un database.
ST_alter_coordsys	Aggiorna l'attributo di un sistema di coordinate nel database.
ST_alter_srs	Aggiorna l'attributo di un sistema di riferimento spaziale nel database.
ST_create_coordsys	Crea un sistema di coordinate nel database.
ST_create_srs	Crea un sistema di riferimento spaziale nel database.
ST_disable_autogeocoding	Indica a DB2 Spatial Extender di interrompere la sincronizzazione di una colonna sottoposta a geocoding con le colonne associate.
ST_disable_db	Elimina le risorse che consentono a DB2 Spatial Extender di memorizzare i dati spaziali e di supportare le operazioni eseguite per questi dati.

Tabella 9. (Continua)

Procedura memorizzata	Descrizione
ST_drop_coordsys	Elimina un sistema di coordinate dal database.
ST_drop_srs	Elimina un sistema di riferimento spaziale dal database.
ST_enable_autogeocoding	Indica a DB2 Spatial Extender di sincronizzare una colonna sottoposta a geocoding con le colonne associate.
ST_enable_db	Fornisce al database le risorse necessarie per memorizzare i dati spaziali e supportare le operazioni relative ai dati.
ST_export_shape	Esporta i dati selezionati nel database in un file shape.
ST_import_shape	Importa un file shape in un database.
ST_register_geocoder	Registra un geocoder diverso da DB2SE_USA_GEOCODER, che è parte del prodotto DB2 Spatial Extender.
ST_register_spatial_column	Registra una colonna spaziale e le associa un sistema di riferimento spaziale.
ST_remove_geocoding_setup	Elimina tutte le informazioni di impostazione delle funzioni di geocoding per la colonna sottoposta a geocoding.
ST_run_geocoding	Esegue il geocoder in modalità batch.
ST_setup_geocoding	Associa una colonna da sottoporre a geocoding a un geocoder ed imposta i valori dei parametri di geocoding corrispondenti.
ST_unregister_geocoder	Annula la registrazione di un geocoder diverso da DB2SE_USA_GEOCODER.
ST_unregister_spatial_column	Annula la registrazione di una colonna spaziale.

Programma di esempio DB2 Spatial Extender

Il programma di esempio DB2[®] Spatial Extender, runGseDemo, può essere utilizzato per verificare l'installazione di DB2 Spatial Extender e per acquisire familiarità con il programma. Per ulteriori informazioni sulla verifica dell'installazione di Spatial Extender, fare riferimento alle "Attività correlate" al termine di questa sezione.

- In ambiente UNIX[®], il programma runGseDemo si trova nel seguente percorso:
\$HOME/sql1lib/samples/spatial

dove \$HOME è la directory principale del proprietario dell'istanza.

- In ambiente Windows[®], il programma runGseDemo si trova nel seguente percorso:
c:\Program Files\IBM\sql1lib\samples\spatial

dove c:\Programmi\IBM\sql1lib è la directory in cui è stato installato DB2 Spatial Extender.

Scrittura di applicazioni ed utilizzo del programma di esempio

Il programma di esempio DB2 Spatial Extender runGseDemo semplifica le operazioni programmazione dell'applicazione. Utilizzando il programma, sarà possibile abilitare un database per le operazioni spaziali in cui verrà eseguita l'analisi spaziale dei dati. Il database conterrà tabelle con informazioni fittizie relative ai clienti e alle zone a rischio. In base a queste informazioni sarà possibile utilizzare Spatial Extender per determinare i clienti che potrebbero essere danneggiati da un'inondazione.

Con il programma di esempio, è possibile:

- Impostare le operazioni richieste per creare e gestire i database abilitati per le operazioni spaziali.
- Richiamare le procedure memorizzate da un programma applicativo.
- Tagliare ed incollare il codice di esempio nelle proprie applicazioni.

Utilizzare il seguente programma di esempio per codificare le attività di DB2 Spatial Extender. Ad esempio, se si desidera scrivere un'applicazione che utilizzi l'interfaccia del database per richiamare le procedure memorizzate di DB2 Spatial Extender, è possibile copiare il codice dal programma di esempio per personalizzare la propria applicazione. Se ancora non si dispone di una certa dimestichezza nel programmare DB2 Spatial Extender, è possibile eseguire il programma di esempio in cui ciascuna operazione viene mostrata in tutti i dettagli. Per ulteriori informazioni sull'esecuzione del programma di esempio, fare riferimento alle "Attività correlate" al termine di questa sezione.

La seguente tabella descrive tutte le operazioni da eseguire nel programma di esempio. In ciascun passo descritto verrà eseguita un'azione e in molti casi verrà indicato anche come annullarla. Ad esempio, la prima operazione consiste nell'abilitare e disabilitare il database spaziale. In tal modo sarà possibile acquisire una certa pratica con la maggior parte delle procedure memorizzate di Spatial Extender. Per ulteriori informazioni sulle procedure memorizzate relative a ciascuna operazione, fare riferimento alle "Attività correlate" al termine di questa sezione.

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender

Operazioni	Azione e descrizione
Abilitare o disabilitare il database spaziale	<ul style="list-style-type: none">• Abilitare il database spaziale E' la prima operazione necessaria per utilizzare DB2 Spatial Extender. Un database abilitato per le operazioni spaziali dispone di una serie di tipi spaziali, una serie di funzioni spaziali, una serie di predicati spaziali, nuovi tipi di indici e una serie di tabelle e viste del catalogo spaziali.• Disabilitare il database spaziale Questa operazione generalmente viene eseguita quando è stato abilitato il database sbagliato, oppure se non si desidera più eseguire le operazioni spaziali in un database precedentemente abilitato. Disabilitando un database spaziale, vengono eliminati le serie di tipi spaziali, di funzioni spaziali, di predicati spaziali, i nuovi tipi di indice e la serie di tabelle e viste del catalogo spaziali associate al database.• Abilitare il database spaziale Come sopra.

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender (Continua)

Operazioni	Azione e descrizione
Creare o eliminare un sistema di coordinate	<ul style="list-style-type: none"> • Creare il sistema di coordinate NORTH_AMERICAN Eseguendo questa operazione, viene creato un nuovo sistema di coordinate nel database. • Cancellare il sistema di coordinate NORTH_AMERICAN Eseguendo questa operazione, il sistema di coordinate NORTH_AMERICAN viene cancellato dal database. • Creare il sistema di coordinate KY_STATE_PLANE Eseguendo questa operazione, viene creato il nuovo sistema di coordinate KY_STATE_PLANE che sarà utilizzato dal sistema di riferimento spaziale creato mediante l'operazione successiva.
Creare o eliminare un sistema di riferimento spaziale	<ul style="list-style-type: none"> • Creare il sistema di riferimento spaziale SRSDEMO1 Mediante questa operazione, viene definito un nuovo sistema di riferimento spaziale (SRS) che sarà utilizzato per interpretare le coordinate. L'SRS contiene dati geometrici che possono essere memorizzati nella colonna di un database abilitato per le operazioni spaziali. Dopo aver registrato l'SRS in una determinata colonna spaziale, sarà possibile memorizzare le coordinate applicabili alla colonna nella colonna associata della tabella CUSTOMERS. • Cancellare l'SRS SRSDEMO1 Eseguire questa operazione se non si desidera utilizzare più l'SRS nel database. Eliminando un SRS, verranno eliminate tutte le relative definizioni dal database. • Creare l' SRS KY_STATE_SRS
Creare le tabelle spaziali ed inserirvi i dati	<ul style="list-style-type: none"> • Creare la tabella CUSTOMERS • Inserire i dati nella tabella CUSTOMERS La tabella CUSTOMERS rappresenta i dati aziendali memorizzati nel database per diversi anni. • Modificare la tabella CUSTOMERS aggiungendovi la colonna LOCATION L'istruzione ALTER TABLE aggiunge una nuova colonna (LOCATION) di tipo ST_Point. I dati verranno inseriti nella colonna eseguendo il geocoding delle colonne di indirizzi come descritto in una delle operazioni successive. • Creare la tabella OFFICES La tabella OFFICES rappresenta, tra gli altri dati, l'area vendite di ciascun ufficio di una compagnia di assicurazioni. Nell'operazione seguente, verranno inseriti i dati di attributo provenienti da un database non DB2 nella tabella. Inoltre verranno importati i dati di attributo nella tabella OFFICES da un file shape.

Scrittura di applicazioni ed utilizzo del programma di esempio

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender (Continua)

Operazioni	Azione e descrizione
Inserire i dati nella colonna	<ul style="list-style-type: none">• Eseguire il geocoding dei dati di indirizzo per la colonna LOCATION della tabella CUSTOMERS utilizzando il geocoder KY_STATE_GC Questa operazione consente di eseguire il geocoding spaziale in modalità batch richiamando l'utilità del geocoder. Il geocoding in modalità batch generalmente viene utilizzato quando è necessario sottoporre a geocoding una parte sostanziosa della tabella.• Individuare la tabella OFFICES creata in precedenza dal file shape utilizzando il sistema di riferimento spaziale KY_STATE_SRS Questa operazione consente di caricare nella tabella OFFICES dati spaziali esistenti nel formato di un file shape. Poiché la tabella OFFICES esiste già, l'utilità LOAD aggiungerà i nuovi record alla tabella esistente.• Creare e caricare la tabella FLOODZONES dal file shape utilizzando il sistema di riferimento spaziale KY_STATE_SRS Questa operazione consente di caricare nella tabella FLOODZONES dati spaziali esistenti nel formato di un file shape. Poiché la tabella non esiste, l'utilità LOAD prima di caricare i dati creerà una tabella.• Creare e caricare la tabella REGIONS dal file shape utilizzando il sistema di riferimento spaziale KY_STATE_SRS
Registrare o annullare la registrazione del geocoder	<ul style="list-style-type: none">• Registrare il geocoder SAMPLEGC• Annullare la registrazione del geocoder SAMPLEGC• Registrare il geocoder KY_STATE_GC <p>Le seguenti operazioni consentono di registrare e di annullare la registrazione del geocoder SAMPLEGC, quindi di creare un nuovo geocoder, KY_STATE_GC, da utilizzare nel programma di esempio.</p>
Creare gli indici spaziali	<ul style="list-style-type: none">• Creare gli indici spaziali per la colonna LOCATION della tabella CUSTOMERS• Eliminare l'indice di griglia spaziale per la colonna LOCATION della tabella CUSTOMERS• Creare gli indici spaziali per la colonna LOCATION della tabella CUSTOMERS• Creare gli indici spaziali per la colonna LOCATION della tabella OFFICES• Creare gli indici spaziali per la colonna LOCATION della tabella FLOODZONES• Creare gli indici spaziali per la colonna LOCATION della tabella REGIONS <p>Queste operazioni consentono di creare l'indice di griglia spaziale per la tabelle CUSTOMERS, OFFICES, FLOODZONES e REGIONS.</p>

Scrittura di applicazioni ed utilizzo del programma di esempio

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender (Continua)

Operazioni	Azione e descrizione
Abilitare la funzione di geocoding automatica	<ul style="list-style-type: none">• Impostare la funzione di geocoding per la colonna LOCATION della tabella CUSTOMERS con il geocoder KY_STATE_GC <p>Questa operazione consente di associare la colonna LOCATION della tabella CUSTOMERS al geocoder KY_STATE_GC e di impostare i valori corrispondenti dei parametri di geocoding.</p> <ul style="list-style-type: none">• Abilitare la funzione di geocoding automatica per la colonna LOCATION della tabella CUSTOMERS <p>Questa operazione consente di attivare il richiamo automatico del geocoder. Utilizzando la funzione automatica di geocoding le colonne LOCATION, STREET, CITY, STATE e ZIP della tabella CUSTOMERS verranno sincronizzate le une alle altre per le successive operazioni di inserimento e aggiornamento dati.</p>
Eseguire le operazioni di inserimento, aggiornamento ed eliminazione nella tabella CUSTOMERS	<ul style="list-style-type: none">• Inserire alcuni record con il nome della via differente• Aggiornare alcuni record con un nuovo indirizzo• Eliminare tutti i record dalla tabella <p>Queste operazioni consentono di inserire, aggiornare ed eliminare i dati nelle colonne STREET, CITY, STATE e ZIP della tabella CUSTOMERS. Dopo aver abilitato la funzione di geocoding automatica, i dati inseriti o aggiornati nelle colonne verranno automaticamente sottoposti a geocoding nella colonna LOCATION. Questo processo è stato abilitato mediante l'operazione precedente.</p>
Disabilitare la funzione di geocoding automatica	<ul style="list-style-type: none">• Disabilitare la funzione di geocoding automatica per la colonna LOCATION nella tabella CUSTOMERS• Eliminare l'impostazione di geocoding per la colonna LOCATION della tabella CUSTOMERS• Eliminare l'indice spaziale per la colonna LOCATION della tabella CUSTOMERS <p>Queste operazioni consentono di disabilitare il richiamo automatico del geocoder e dell'indice spaziale come preparazione per l'operazione successiva, che consentirà di sottoporre nuovamente a geocoding tutti i dati della tabella CUSTOMERS.</p> <p>Suggerimento: Se si desidera caricare una grande quantità di geodati, eliminare l'indice spaziale prima di caricare i dati, quindi crearlo nuovamente dopo aver eseguito l'operazione.</p>

Scrittura di applicazioni ed utilizzo del programma di esempio

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender (Continua)

Operazioni	Azione e descrizione
Sottoporre nuovamente a geocoding la tabella CUSTOMERS	<ul style="list-style-type: none">• Sottoporre nuovamente a geocoding la colonna LOCATION della tabella CUSTOMERS con un livello di precisione inferiore: 90% invece di 100%• Ricreare l'indice spaziale per la colonna LOCATION della tabella CUSTOMERS• Abilitare nuovamente la funzione di geocoding automatica con un livello di precisione inferiore: 90% invece di 100% <p>Queste operazioni consentono di eseguire il geocoder in modalità batch, creare nuovamente l'indice spaziale e riabilitare la funzione di geocoding automatica con un nuovo livello di precisione. L'amministratore spaziale dovrebbe eseguire l'azione se durante il processo di geocoding si verifica un alto numero di errori. Se il livello di precisione è impostato su 100%, è possibile che l'operazione di geocoding abbia esito negativo in quanto non viene rilevato alcun indirizzo corrispondente al 100% ai dati di riferimento. Diminuendo il livello, il geocoder ha più possibilità di individuare dei dati. Dopo aver eseguito nuovamente il geocoding della tabella in modalità batch, la funzione di geocoding automatica viene riabilitata e l'indice spaziale ricreato. In tal modo si consente di effettuare una gestione incrementale dell'indice e della colonna spaziali per le successive operazioni di inserimento e aggiornamento.</p>
Creare una vista e registrare la colonna spaziale nella vista	<ul style="list-style-type: none">• Creare la vista HIGHRISKCUSTOMERS, basata sull'unione delle tabelle CUSTOMERS e FLOODZONES• Registrare la colonna spaziale della vista <p>Queste operazioni consentono di creare una vista e di registrarne la colonna spaziale.</p>
Eseguire analisi spaziali	<ul style="list-style-type: none">• Individuare il numero di clienti serviti da ciascuna regione (ST_Within)• Per uffici e clienti situati nella stessa regione, individuare il numero dei clienti che si trovano entro una determinata distanza da ciascun ufficio (ST_Within, ST_Distance)• Per ciascuna regione, individuare il reddito medio ed il premio di ciascun cliente (ST_Within)• Individuare il numero delle zone a rischio a cui si sovrappongono le aree in cui si trovano gli uffici (ST_Overlaps)• Individuare l'ufficio più vicino alla posizione del cliente presupponendo che l'ufficio si trovi nella parte centrale della zona uffici (ST_Distance)• Individuare i clienti le cui posizioni sono vicine ai confini di una determinata zona a rischio (ST_Buffer, ST_Intersects)• Individuare i clienti ad alto rischio entro una certa distanza da un determinato ufficio (ST_Within) <p>Tutte queste operazioni utilizzano la procedura memorizzata sqlRunSpatialQueries.</p> <p>Viene eseguita l'analisi spaziale utilizzando i predicati e le funzioni spaziali di DB2 SQL. Per migliorare le prestazioni dell'interrogazione laddove possibile, l'utilità di ottimizzazione DB2 si servirà dell'indice spaziale per le colonne.</p>

Tabella 10. Operazioni del programma di esempio di DB2 Spatial Extender (Continua)

Operazioni	Azione e descrizione
Esportazione dei dati spaziali in file shape	<ul style="list-style-type: none">• Esportare la vista HIGHRISKCUSTOMERS nei file shape <p>Questa operazione consente di esportare la vista HIGHRISKCUSTOMERS nei file shape. Esportando i dati dal formato di un database in un altro formato file, le informazioni potranno essere utilizzate da altri strumenti (ad esempio, ArcExplorer for DB2).</p> <p>L'operazione è compresa nel programma runGseDemo.c e viene riportata solo come riferimento. E' possibile modificare il programma di esempio per indicare l'ubicazione del file shape di esportazione e eseguire nuovamente il programma di esempio.</p>
Esportare ed importare i file SDE	<ul style="list-style-type: none">• Esportare la tabella CUSTOMERS in un file di trasferimento SDE• Importare i dati dal file di trasferimento SDE appena esportato <p>Queste operazioni rappresentano un esempio di importazione ed esportazione dei file di trasferimento SDE.</p> <p>Le operazioni sono comprese nel programma runGseDemo.c e vengono riportate solo come riferimento. E' possibile modificare il programma di esempio per indicare l'ubicazione del file shape di esportazione e eseguire nuovamente il programma di esempio.</p>

Attività correlate:

- "Verifica dell'installazione di Spatial Extender" a pagina 39
- "Risoluzione dei problemi di installazione" a pagina 40
- "Scrittura di applicazioni per DB2 Spatial Extender" a pagina 133
- "Richiamo delle procedure memorizzate di DB2 Spatial Extender da un'applicazione" a pagina 134
- "Inserimento del file di installazione di DB2 Spatial Extender nelle applicazioni spaziali" a pagina 133

Capitolo 15. Individuazione dei problemi con DB2 Spatial Extender

Se si verifica un problema durante l'utilizzo di DB2 Spatial Extender, è possibile stabilirne le cause utilizzando DB2 Spatial Extender come indicato di seguito:

- Per diagnosticare il problema, è possibile utilizzare le informazioni contenute nei messaggi.
- Quando vengono utilizzate le procedure memorizzate e le funzioni di Spatial Extender, DB2 restituisce informazioni sull'esito delle operazioni eseguite. Le informazioni vengono restituite in un codice di messaggio (in formato di numero intero), in un messaggio di testo o entrambi, in base all'interfaccia utilizzata per lavorare con DB2 Spatial Extender.
- E' possibile visualizzare il file di notifica per la gestione DB2, in cui vengono registrate informazioni di diagnostica relative agli errori.
- Se in Spatial Extender si verifica un problema ricorrente e riproducibile, il rappresentante dell'assistenza tecnica di IBM potrebbe richiedere all'utente di utilizzare la funzione di traccia DB2 per raccogliere informazioni utili per diagnosticare il problema.

In questo capitolo vengono trattati questi argomenti.

Interpretazione dei messaggi di DB2 Spatial Extender

DB2[®] Spatial Extender può essere utilizzato mediante quattro diverse interfacce:

- Procedure memorizzate DB2 Spatial Extender
- Funzioni DB2 Spatial Extender
- CLP (command line processor) DB2 Spatial Extender
- Centro di controllo DB2

Tutte le interfacce restituiscono messaggi di DB2 Spatial Extender che indicano se l'operazione spaziale richiesta è stata eseguita correttamente o se si sono verificati degli errori.

La seguente tabella spiega il contenuto delle parti che compongono il seguente messaggio di errore di esempio di DB2 Spatial Extender:

GSE0000I: Operazione eseguita correttamente.

Tabella 11. Parti del messaggio di DB2 Spatial Extender

Parte del messaggio	Descrizione
GSE	Identificatore del messaggio. Tutti i messaggi di DB2 Spatial Extender iniziano con il prefisso GSE.
0000	Numero del messaggio. Un numero di quattro cifre che può variare da 0000 a 9999.

Individuazione dei problemi

Tabella 11. Parti del messaggio di DB2 Spatial Extender (Continua)

Parte del messaggio	Descrizione
I	Tipo di messaggio. Una lettera singola che indica il livello di gravità del messaggio: C Messaggio di errore critico N Messaggio di errore non critico W Messaggi di avviso I Messaggi informativi
Operazione eseguita correttamente.	Spiegazione del messaggio.

Il testo del messaggio contiene una breve spiegazione del problema. Per richiamare altre informazioni che comprendono dettagli e suggerimenti per evitare o correggere il problema, procedere come segue:

1. Aprire un prompt dei comandi del sistema operativo.
2. Immettere il comandi di aiuto DB2 riportando l'identificativo e il numero del messaggio in modo da visualizzare tutte le informazioni possibili. Ad esempio:
DB2 "? GSEnnnn"

dove *nnnn* rappresenta il numero del messaggio.

E' possibile immettere l'identificativo del messaggio e la lettera che indica il tipo di messaggio in caratteri maiuscoli o minuscoli. I comandi DB2 "? GSE0000I" e db2 "? gse0000i", otterranno lo stesso risultato.

E' possibile omettere la lettera che segue il numero del messaggio. Ad esempio, il comando DB2 "? GSE0000" otterrà lo stesso risultato del comando DB2 "? GSE0000I".

Se il codice del comando è GSE4107N, immettendo DB2 "? GSE4107N" nel prompt dei comandi, verranno visualizzate le seguenti informazioni:

GSE4107N Il valore dimensione griglia "<dimensione-griglia>" non è valido nella posizione in cui è stato utilizzato.

Spiegazione: La dimensione della griglia specificata "<dimensione-griglia>" non è valida.

Durante la creazione dell'indice di griglia utilizzando l'istruzione CREATE INDEX, è stata effettuata una delle seguenti specifiche non valide:

- E' stato specificato un numero minore di 0 (zero) come dimensione griglia per il primo, secondo o terzo livello.
- E' stato specificato il valore 0 (zero) come dimensione griglia per il primo livello.
- La dimensione griglia specificata per il secondo livello è inferiore alla dimensione del primo livello ma non è uguale a 0 (zero).
- La dimensione griglia specificata per il terzo livello è inferiore alla dimensione del secondo livello ma non è uguale a 0 (zero).
- La dimensione griglia specificata per il terzo livello è maggiore di 0 (zero) ma la dimensione specificata per il secondo livello è 0 (zero).

Risposta utente: Indicare un valore valido per la dimensione della griglia.

codicemsg: -4107

statusql: 38SC7

Se le informazioni sono troppe per essere visualizzate in un solo pannello e il sistema operativo utilizzato supporta l'eseguibile e le pipe **more**, immettere il seguente comando:

```
db2 "? GSEnnn" | more
```

Il programma **more** farà sì che la visualizzazione venga interrotta dopo ciascun pannello, in modo che sia possibile leggere le informazioni.

Argomenti correlati:

- "Parametri di output delle procedure memorizzate di DB2 Spatial Extender" a pagina 145
- "Messaggi delle funzioni di DB2 Spatial Extender" a pagina 147
- "Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender" a pagina 149
- "Messaggi del Centro di controllo DB2" a pagina 151
- "File di notifica per la gestione" a pagina 153

Attività correlate:

- "Registrazione dei problemi di DB2 Spatial Extender mediante il comando db2trc" a pagina 152

Riferimenti correlati:

- "GSE messages" nella *Message Reference Volume 1*

Parametri di output delle procedure memorizzate di DB2 Spatial Extender

Le procedure memorizzate di DB2[®] Spatial Extender vengono richiamate *implicitamente* quando Spatial Extender viene utilizzato dal Centro di controllo DB2 e dal CLP di DB2 Spatial Extender (db2se). E' possibile richiamare le procedure memorizzate *esplicitamente* in un programma applicativo o dalla riga comandi DB2.

Questa sezione descrive come diagnosticare eventuali problemi che si verificano richiamando le procedure memorizzate esplicitamente in programmi applicativi o dalla riga comandi DB2. Per diagnosticare le procedure memorizzate richiamate implicitamente, utilizzare i messaggi restituiti dal CLP di DB2 Spatial Extender o dal Centro di controllo DB2. Questi messaggi verranno discussi in una diversa sezione.

Le procedure memorizzate di DB2 Spatial Extender hanno due parametri di output: il codice del messaggio (codice_msg) e il testo del messaggio (testo_msg). I valori dei parametri indicano l'esito positivo o negativo di una procedura memorizzata.

codice_msg

Il parametro codice_msg è costituito da un numero intero positivo, negativo o uguale a zero (0). I numeri positivi sono utilizzati per indicare avvertenza, i negativi per gli errori (sia critici che non) e zero (0) per i messaggi informativi.

Individuazione dei problemi

Il valore assoluto del codice_msg è incluso nel parametro testo_msg come numero del messaggio. Ad esempio:

- Se il codice_msg è 0, il numero del messaggio sarà 0000.
- Se il codice_msg è -219, il numero del messaggio sarà 0219. Un codice_msg negativo, indica che il messaggio si riferisce a un errore critico o non critico.
- Se il codice_msg è +1036, il numero del messaggio sarà 1036. Un numero di codice_msg positivo indica che il messaggio contiene un'avvertenza.

I numeri di codice_msg per le procedure memorizzate di Spatial Extender sono divisi in tre categorie, contenute nella seguente tabella:

Tabella 12. Codici messaggio delle procedure memorizzate

Codici	Categoria
0000 – 0999	Messaggi comuni
1000 – 1999	Messaggi di gestione
2000 – 2999	Messaggi di importazione ed esportazione

testo_msg

Il parametro testo_msg contiene l'identificativo, il numero, il tipo di messaggio e una spiegazione. Un esempio di testo_msg di una procedura memorizzata è:

```
GSE0219N Un'istruzione EXECUTE IMMEDIATE
non riuscita. SQLERROR = "<errore-sql>".
```

Il parametro testo_msg rappresenta una breve spiegazione del problema. E' possibile richiamare altre informazioni che comprendono dettagli e suggerimenti per evitare o correggere il problema.

Per una spiegazione dettagliata delle parti che compongono il parametro testo_msg e per le istruzioni su come richiamare ulteriori informazioni sul messaggio, consultare la sezione Interpretazione dei messaggi di DB2 Spatial Extender.

Utilizzo delle procedure memorizzate nelle applicazioni:

Richiamando una procedura memorizzata DB2 Spatial Extender da un'applicazione, vengono ricevuti i parametri di output codice_msg e testo_msg. E' possibile:

- Programmare l'applicazione in modo che restituisca i valori dei parametri di output all'utente dell'applicazione.
- Svolgere alcune operazioni in base al tipo di valore codice_msg restituito.

Utilizzo delle procedure memorizzate da una riga comandi DB2:

Richiamando una procedura memorizzata DB2 Spatial Extender da una riga comandi DB2 vengono ricevuti i parametri di output codice_msg e testo_msg. I valori dei parametri indicano l'esito positivo o negativo della procedura memorizzata.

Se si stabilisce una connessione al database e si desidera richiamare la procedura ST_disable_db, utilizzare un comando DB2 CALL, come mostrato nell'esempio

seguito, in modo da disabilitare le operazioni spaziali e visualizzare i risultati dei valori di output. Viene utilizzato il parametro `force` di valore 0, e due punti interrogativi al termine del comando `CALL` per rappresentare i parametri di output `codice_msg` e `testo_msg`. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

```
call db2gse.st_disable_db(0, ?, ?)
```

Valori dei parametri di output

```
-----
Nome parametro : MSGCODE
Valore parametro: 0
```

```
Nome parametro : MSGTEXT
Valore parametro: GSE0000I Operazione eseguita correttamente.
```

```
Stato restituzione = 0
```

Se il `testo_msg` restituito è `GSE2110N`, utilizzare il comando di aiuto `DB2` per visualizzare più informazioni sul messaggio. Ad esempio:

```
"? GSE2110"
```

Vengono visualizzate le seguenti informazioni:

```
GSE2110N    Il sistema di riferimento spaziale per la
             forma geometrica nella riga "<numero-riga>" non è valido.
             L'identificativo numerico del sistema di riferimento spaziale
             è "<srs-id>".
```

Spiegazione: Alla riga *numero-riga*, la forma geometrica da esportare utilizza un sistema di riferimento spaziale non valido. Impossibile esportare la forma geometrica.

Risposta utente: Correggere la forma geometrica indicata o escludere la riga dall'operazione di esportazione modificando l'istruzione `SELECT`.

```
codice_msg: -2110
```

```
statusql: 38S9A
```

Argomenti correlati:

- “Interpretazione dei messaggi di DB2 Spatial Extender” a pagina 143
- “Messaggi delle funzioni di DB2 Spatial Extender” a pagina 147
- “Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender” a pagina 149
- “Messaggi del Centro di controllo DB2” a pagina 151

Riferimenti correlati:

- “GSE messages” nella *Message Reference Volume 1*

Messaggi delle funzioni di DB2 Spatial Extender

I messaggi restituiti dalle funzioni DB2® Spatial Extender, generalmente vengono inseriti in un messaggio SQL. Il `SQLCODE` restituito nel messaggio indica se si è verificato un errore o se esiste un'avvertenza associata alla funzione. Ad esempio:

- L'`SQLCODE -443` (numero messaggio `SQL0443`) indica che si è verificato un errore con la funzione.

Individuazione dei problemi

- L'SQLCODE +462 (numero messaggio SQL0462) indica che esiste un avvertenza associata alla funzione.

Nella tabella è riportata una spiegazione delle parti più importanti del seguente messaggio di esempio:

```
DB21034E Il comando è stato elaborato come istruzione SQL perché non era
un comando CLP valido. Durante l'elaborazione SQL, ha restituito:
SQL0443N La routine "DB2GSE.GSEGEOMFROMWKT"
(nome specifico "GSEGEOMWKT1") ha restituito un errore
SQLSTATE con testo di diagnostica "GSE3421N Poligono non chiuso.".
SQLSTATE=38SSL
```

Tabella 13. Parti principali dei messaggi delle funzioni di DB2 Spatial Extender

Parte del messaggio	Descrizione
SQL0443N	L'SQLCODE indica il tipo di problema.
GSE3421N	Numero di messaggi di DB2 Spatial Extender e tipo di messaggi. I numeri dei messaggi relativi alle funzioni sono compresi tra GSE3000 e GSE3999. Inoltre, è possibile che durante lo svolgimento delle operazioni con DB2 Spatial Extender vengano restituiti messaggi comuni. I numeri dei messaggi comuni, sono compresi tra GSE0001 e GSE0999.
Poligono non chiuso	Spiegazione del messaggio di DB2 Spatial Extender.
SQLSTATE=38SSL	Un codice SQLSTATE che identifica ulteriormente l'errore. Viene restituito un codice SQLSTATE per ciascuna istruzione o riga. <ul style="list-style-type: none">• I codici SQLSTATE per gli errori relativi alle funzioni di Spatial Extender sono 38Sxx, dove ciascuna x corrisponderà a una lettera o a un numero.• I codici SQLSTATE per le avvertenze relative alle funzioni di Spatial Extender sono 01HSx, dove x corrisponderà a una lettera o a un numero.

Esempio di messaggio di errore SQL0443:

L'utente tenta di inserire i valori relativi a un poligono nella tabella POLYGON_TABLE, come mostrato di seguito:

```
INSERT INTO polygon_table ( geometry )
VALUES ( ST_Polygon ( 'polygon (( 0 0, 0 2, 2 2, 1 2)) ' ) )
```

Viene ricevuto un messaggio di errore perché non è stato fornito il valore finale per chiudere il programma. Il messaggio di errore restituito è:

```
DB21034E Il comando è stato elaborato come istruzione SQL perché non era
un comando CLP valido. Durante l'elaborazione SQL, ha restituito:
SQL0443N La routine "DB2GSE.GSEGEOMFROMWKT"
(nome specifico "GSEGEOMWKT1") ha restituito un errore
SQLSTATE con testo di diagnostica "GSE3421N Poligono non chiuso.".
SQLSTATE=38SSL
```

Il numero del messaggio SQL SQL0443N indica che si è verificato un errore, e il testo del messaggio Spatial Extender riporta GSE3421N Poligono non chiuso.

Se viene ricevuto questo tipo di messaggio:

1. Individuare il numero di messaggio GSE nel messaggio di errore DB2 o SQL.

- Utilizzare il comando di aiuto DB2 (DB2 ?) per visualizzare la spiegazione del messaggio di Spatial Extender e la risposta utente. Facendo riferimento all'esempio precedente, immettere il seguente comando nel prompt dei comandi del sistema operativo:

```
DB2 "? GSE3421"
```

Il messaggio viene visualizzato nuovamente, insieme ad una spiegazione dettagliata e alla risposta utente consigliata.

Argomenti correlati:

- “Interpretazione dei messaggi di DB2 Spatial Extender” a pagina 143
- “Parametri di output delle procedure memorizzate di DB2 Spatial Extender” a pagina 145
- “Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender” a pagina 149
- “Messaggi del Centro di controllo DB2” a pagina 151

Riferimenti correlati:

- “GSE messages” nella *Message Reference Volume 1*

Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender

Il processore riga comandi (db2se) di DB2[®] Spatial Extender, restituisce messaggi per:

- Procedure memorizzate, se richiamate implicitamente.
- Informazioni di forma, se dal processore di riga comandi di DB2 Spatial Extender, è stato richiamato il programma secondario **shape_info**. Si tratta di messaggi informativi.
- Operazioni di migrazione.
- Operazioni di importazione ed esportazione di dati shape in e dal client.

Esempi di messaggi di procedure memorizzate restituiti dal processore riga comandi di DB2 Spatial Extender:

La maggior parte dei messaggi restituiti mediante il processore riga comandi di DB2 Spatial Extender riguardano le procedure memorizzate di DB2 Spatial Extender. Richiamando una procedura memorizzata dal processore riga comandi di DB2 Spatial Extender, viene ricevuto un messaggio che indica se la procedura memorizzata ha avuto esito negativo o positivo.

Il testo del messaggio contiene l'identificativo, il numero e il tipo di messaggio e una spiegazione. Ad esempio, dopo aver abilitato un database mediante il comando `db2se enable_db testdb`, il testo del messaggio restituito dal processore riga comandi di Spatial Extender sarà simile a quello riportato di seguito:

Abilitazione del database in corso. Attendere...

```
GSE1036W  Operazione riuscita. I
          valori per la gestione del database e dei
          parametri di configurazione del database
          dovrebbero essere aumentati.
```

Individuazione dei problemi

Allo stesso modo, dopo aver abilitato un database mediante il comando db2se disable_db testdb, il testo del messaggio restituito dal processore riga comandi di Spatial Extender sarà simile a quello riportato di seguito:

```
GSE0000I Operazione eseguita correttamente.
```

Il testo del messaggio contiene una breve spiegazione del problema. Per richiamare altre informazioni che comprendono dettagli e suggerimenti per evitare o correggere il problema, fare riferimento all'apposita sezione.

Se si desidera richiamare le procedure memorizzate mediante un programma applicativo o dalla riga comandi DB2, fare riferimento alla sezione relativa ai parametri di output di diagnostica.

Esempio di messaggi contenenti informazioni di tipo shape restituiti dal processore di riga comandi di Spatial Extender:

Se si desidera visualizzare le informazioni su file shape, ad esempio il file office, immettere il seguente comando nel processore di riga comandi (db2se) di Spatial Extender:

```
db2se shape_info -fileName /tmp/offices
```

Di seguito è riportato un esempio delle informazioni che verranno visualizzate:

Informazioni sul file shape

```
-----  
Codice file = 9994  
Lunghezza file (parole a 16-bit) = 484  
Versione file shape = 1000  
Tipo shape = 1 (ST_POINT)  
Numero di record = 31
```

```
Coordinata X minima = -87.053834  
Coordinata X massima = -83.408752  
Coordinata Y minima = 36.939628  
Coordinata Y massima = 39.016477  
Le forme non hanno coordinate Z.  
Le forme non hanno coordinate M.
```

E' stato rilevato il file shape di indice (estensione .shx).

Informazioni sul file di attributi

```
-----  
Codice file dBase = 3  
Data dell'ultimo aggiornamento = 1901-08-15  
Numero di record = 31  
Numero di byte nell'intestazione = 129  
Numero di byte in ciascun record = 39  
Numero di colonne = 3
```

Numero colonna	Nome colonna	Tipo di dati	Lunghezza	Decimale
1	NAME	C (Carattere)	16	0
2	EMPLOYEES	N (Numeric)	11	0
3	ID	N (Numerico)	11	0

```
Definizione sistema di coordinate: "GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]"
```

Esempi di messaggi sulla migrazione restituiti dal processore di riga comandi di Spatial Extender:

Dopo aver richiamato i comandi per l'esecuzione della migrazione, vengono restituiti dei messaggi che indicano l'esito positivo o negativo dell'operazione.

Se si avvia la migrazione del database miodb mediante il comando `db2se migrate miodb -messagesFile /tmp/migrate.msg`, Il testo del messaggio restituito dal processore di riga comandi di Spatial Extender sarà simile a quello riportato di seguito:

```
Migrazione del database in corso. Attendere...  
GSE0000I  Operazione eseguita correttamente.
```

Argomenti correlati:

- "Interpretazione dei messaggi di DB2 Spatial Extender" a pagina 143
- "Parametri di output delle procedure memorizzate di DB2 Spatial Extender" a pagina 145
- "Messaggi delle funzioni di DB2 Spatial Extender" a pagina 147
- "Messaggi del Centro di controllo DB2" a pagina 151

Riferimenti correlati:

- "GSE messages" nella *Message Reference Volume 1*

Messaggi del Centro di controllo DB2

Quando DB2® Spatial Extender viene utilizzato attraverso il Centro di controllo DB2, nella finestra Messaggio DB2, viene visualizzata una serie di messaggi. La maggior parte dei messaggi visualizzati provengono da DB2 Spatial Extender. E' possibile che vengano visualizzati messaggi SQL. I messaggi SQL vengono restituiti quando gli errori riguardano licenze, blocchi o quando un servizio DAS non risulta disponibile. Le seguenti sezioni contengono alcuni esempi di messaggi di DB2 Spatial Extender e di SQL visualizzati nel Centro di controllo DB2.

Messaggi di DB2 Spatial Extender:

Quando si riceve un messaggio di DB2 Spatial Extender attraverso il Centro di controllo, il testo completo del messaggio viene visualizzato nell'area di testo della finestra Messaggio DB2, ad esempio:

```
GSE0219N  Istruzione EXECUTE IMMEDIATE  
          non riuscita. SQLERROR = "<errore-sql>".
```

Messaggi di SQL:

Quando si riceve un messaggio di SQL relativo a DB2 Spatial Extender attraverso il Centro di controllo:

- L'identificativo, il numero e il tipo di messaggio vengono visualizzati nella parte sinistra della finestra Messaggio DB2, ad esempio: SQL0612N.
- Il testo del messaggio viene visualizzato nell'area di testo nella finestra Messaggio DB2.

Il testo del messaggio che viene visualizzato nella finestra Messaggio DB2 potrebbe contenere il testo del messaggio SQL e l'SQLSTATE, oppure il testo del messaggio, una spiegazione dettagliata e la risposta utente.

Un esempio di messaggio SQL contenete il testo del messaggio SQL e l'SQLSTATE è:

Individuazione dei problemi

[IBM][CLI Driver][DB2/NT] SQL0612N "<name>" è un nome duplicato. SQLSTATE=42711

Un esempio di messaggio SQL contenete il testo del messaggio SQL, la spiegazione dettagliata e la risposta utente è:

SQL8008N

Il prodotto "DB2 Spatial Extender" non dispone di una chiave di licenza valida installata e il periodo di valutazione è scaduto.

Spiegazione:

Non è stato possibile individuare una chiave di licenza valida e il periodo di valutazione è scaduto.

Risposta utente:

Installare una chiave di licenza valida per la versione del prodotto. Per ottenere una chiave di licenza del prodotto rivolgersi al rappresentante IBM® o al fornitore autorizzato.

Argomenti correlati:

- "Interpretazione dei messaggi di DB2 Spatial Extender" a pagina 143
- "Parametri di output delle procedure memorizzate di DB2 Spatial Extender" a pagina 145
- "Messaggi delle funzioni di DB2 Spatial Extender" a pagina 147
- "Messaggi del processore di riga comandi (CLP) di DB2 Spatial Extender" a pagina 149

Riferimenti correlati:

- "GSE messages" nella *Message Reference Volume 1*

Registrazione dei problemi di DB2 Spatial Extender mediante il comando db2trc

Se in DB2 Spatial Extender si verifica un problema ricorrente e riproducibile, è possibile utilizzare la funzione di traccia di DB2 per registrare le informazioni sul problema. La funzione viene attivata mediante il comando **db2trc** ed è in grado di:

- Tracciare gli eventi
- Inserire i dati relativi alla traccia in un file
- Formattare i dati relativi alla traccia in un formato leggibile

Limitazioni:

Attivare la funzione di traccia solo con l'aiuto di un rappresentante del supporto tecnico di DB2.

In sistemi operativi UNIX, per registrare la traccia di un'istanza DB2, è necessario disporre di autorizzazione SYSADM, SYSCTRL o SYSMAINT.

In sistemi operativi Windows, non è richiesta nessuna particolare autorizzazione.

Procedura:

Per registrare gli eventi di DB2 Spatial Extender nella memoria, procedere come segue:

1. Chiudere tutte le altre applicazioni.

- Attivare la funzione di traccia. Il rappresentante del supporto tecnico di DB2 fornirà i parametri necessari per eseguire questa operazione. Il comando di base è:

```
db2trc on
```

Limitazioni: È necessario immettere il comando **db2trc** in un prompt dei comandi del sistema operativo o in uno script della shell. Non può essere utilizzato nella riga comando dell'interfaccia di DB2 Spatial Extender (db2se) o nel processore di riga comandi di DB2.

E' possibile registrare le informazioni in memoria o in un file. Il metodo preferito è la registrazione della traccia in memoria. Se il problema ricreato interrompe le attività della stazione di lavoro e non rende possibile recuperare la traccia, registrare le informazioni in un file.

- Riprodurre il problema.
- Inserire i dati di traccia in un file. Ad esempio:

```
db2trc dump january23trace.dmp
```

Questo comando creerà il file *january23trace.dmp* nella directory corrente specificata, e vi inserisce le informazioni relative alla traccia.

E' possibile specificare una directory diversa aggiungendo al comando il percorso del file. Ad esempio, per inserire il file delle informazioni nella directory */tmp/spatial/errors*, la sintassi sarà:

```
db2trc dump /tmp/spatial/errors/january23trace.dmp
```

La traccia deve essere registrata subito dopo il verificarsi del problema.

- Disattivare la funzione di traccia. Ad esempio:

```
db2trc off
```

- Formattare i dati come file ASCII. E' possibile ordinare i dati in due modi:

- L'opzione **flw** ordinerà i dati per processo o per thread. Ad esempio:

```
db2trc flw january23trace.dmp january23trace.flw
```

- L'opzione **fmt** elencherà gli eventi in ordine cronologico. Ad esempio:

```
db2trc fmt january23trace.dmp january23trace.fmt
```

Argomenti correlati:

- "DB2 trace (db2trc)" nella pubblicazione *Troubleshooting Guide*
- "Interpretazione dei messaggi di DB2 Spatial Extender" a pagina 143
- "File di notifica per la gestione" a pagina 153

Riferimenti correlati:

- "GSE messages" nella *Message Reference Volume 1*

File di notifica per la gestione

Le informazioni di diagnostica sugli errori vengono registrate nel file di notifica di gestione. Tali informazioni vengono utilizzate per l'individuazione dei problemi e sono rivolte al supporto tecnico di DB2®.

Il file di notifiche di gestione contiene le informazioni in formato testo registrate da DB2 e da DB2 Spatial Extender. Si trova nella directory specificata dal parametro di configurazione di database manager *DIAGPATH*. Nei sistemi Windows® NT, Windows 2000 e Windows XP, il file di notifiche di gestione DB2 si trova nel file di registrazione eventi e può essere visualizzato con il Visualizzatore eventi di Windows.

Individuazione dei problemi

Le informazioni registrate da DB2 nel file di registrazione di gestione sono determinate dalle impostazioni DIAGLEVEL e NOTIFYLEVEL.

Per visualizzare il file sul computer in cui si è verificato un problema, utilizzare un editor di testo. Gli eventi più recenti sono registrati nella parte finale del file. Generalmente, ogni voce è costituita dalle seguenti parti:

- Una indicazione data/ora.
- Il percorso che contiene l'errore. Gli identificatori dell'applicazione consentono di associare le voci corrispondenti ad un'applicazione nei file di registrazione dei server e dei client.
- Un messaggio di diagnostica che spiega l'errore (generalmente inizia per "DIA" o "ADM").
- Tutti i dati di supporto disponibili, ad esempio strutture di dati SQLCA e indicatori del percorso di altri eventuali file di dump o di registrazione.

Se il database funziona normalmente, queste informazioni possono essere ignorate.

Il file di notifica di gestione cresce continuamente. Se le dimensioni diventano eccessive, effettuare una copia di riserva del file, quindi eliminarlo. Quando il sistema avrà bisogno di tale file, ne verrà generato automaticamente uno nuovo.

Argomenti correlati:

- "Interpreting the administration logs" nella pubblicazione *Troubleshooting Guide*
- "Interpretazione dei messaggi di DB2 Spatial Extender" a pagina 143

Attività correlate:

- "Registrazione dei problemi di DB2 Spatial Extender mediante il comando db2trc" a pagina 152

Riferimenti correlati:

- "GSE messages" nella *Message Reference Volume 1*

Parte 4. Utilizzo di DB2 Geodetic Extender

Capitolo 16. DB2 Geodetic Extender

Questo capitolo introduce DB2 Geodetic Extender, descrivendone la funzionalità, illustra quando utilizzarlo e spiega alcuni concetti geodetici.

DB2 Geodetic Extender

DB2® Geodetic Extender consente di considerare la Terra come un globo. Utilizzando lo stesso tipo di dati spaziali e di funzioni come per le altre operazioni di Spatial Extender, è possibile utilizzare Geodetic Extender per eseguire interrogazioni dei dati vicini ai poli o che attraversano il 180° meridiano. E' possibile conservare i dati che fanno riferimento a un punto preciso sulla superficie della Terra.

Il nome Geodetic Extender deriva dalla disciplina nota come *geodesia*. La geodesia è lo studio delle dimensioni e della forma della Terra (o di qualsiasi corpo avente forma di un ellissoide, ad esempio il Sole o la sfera celeste). Geodetic Extender è progettato per gestire gli oggetti definiti sulla superficie della Terra con un altro livello di precisione.

Per ottenere questo livello di precisione, Geodetic Extender utilizza un sistema di coordinate di latitudine e di longitudine su un modello ellissoidale o *datum geodetico*, invece che un sistema di coordinate piane, con assi *x* e *y*. Un modello ellissoidale evita le distorsioni, le inaccuranze e le imprecisioni che potrebbero presentarsi utilizzando proiezioni piane. Per ulteriori informazioni, consultare le sezioni "Latitudine e longitudine geodetiche" a pagina 159, "Sistema di coordinate geografiche" a pagina 59 e "Sistemi di coordinate proiettate" a pagina 65.

Per accedere alle operazioni geodetiche, invece che alle spaziali, è necessario definire un sistema di riferimento spaziale geodetico per i dati. Lo SRID (spatial reference system ID) di questi sistemi è compreso nell'intervallo 2000000000 - 2000001000. Geodetic Extender fornisce 318 sistemi di riferimento spaziali geodetici predefiniti.

Per poter utilizzare DB2 Geodetic Extender, è necessario installare DB2 Spatial Extender. DB2 Geodetic Extender viene ordinato separatamente da DB2 Spatial Extender, ed è necessario acquistare una licenza separata.

Argomenti correlati:

- "Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender" a pagina 158
- "Datum geodetici" a pagina 158

Attività correlate:

- "Impostazione ed abilitazione di DB2 Geodetic Extender" a pagina 165

Riferimenti correlati:

- "Datum supportati da DB2 Geodetic Extender" a pagina 211

Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender

DB2[®] Spatial Extender e DB2 Geodetic Extender gestiscono entrambi sistemi di informazioni geografici (GIS) in un database DB2. Ciascun extender utilizza diverse tecnologie di base che risolvono problemi diversi e si completano una con l'altra.

- Geodetic Extender considera la Terra come un globo. Utilizza il sistema di coordinate di latitudine e di longitudine su un modello ellissoidale. Le operazioni geometriche sono precise, a prescindere dalla posizione. E' generato su una libreria Hipparchus, fornita su licenza dalla Geodyssey Limited. Per ulteriori informazioni, visitare il sito <http://www.geodyssey.com>.

Si consiglia di utilizzare Geodetic Extender per insiemi di dati globali e applicazioni che coprono una vasta area sulla Terra, laddove una singola proiezione di una carta geografica non potrebbe fornire la precisione richiesta dall'applicazione.

- Spatial Extender considera la Terra come una carta geografica piana. Utilizza forme geometriche planimetriche (piane), il che significa che approssima la superficie curva della Terra proiettandola su una superficie piana. Questa proiezione causa distorsioni, che possono variare con l'estensione dei dati, e che generalmente aumentano verso i limiti della zona proiettata. Tutte le carte geografiche piane sono distorte. Spatial Extender è generato sulla libreria shape ESRI, fornita su licenza da ESRI. Per ulteriori informazioni, visitare il sito <http://www.esri.com>.

Si consiglia di utilizzare Spatial Extender per insiemi di dati locali e regionali ben rappresentati in coordinate proiettate, e per applicazioni in cui la precisione della posizione non è importante. Ad esempio, se una compagnia di assicurazioni medica, desidera sapere la posizione degli ospedali e delle cliniche in una regione o provincia.

Argomenti correlati:

- "DB2 Geodetic Extender" a pagina 157
- "Regioni geodetiche" a pagina 161
- "Latitudine e longitudine geodetiche" a pagina 159
- "Distanze geodesiche" a pagina 160
- "Sferoidi geodetici" a pagina 220

Attività correlate:

- "Impostazione ed abilitazione di DB2 Geodetic Extender" a pagina 165

Datum geodetici

Un datum geodetico è un sistema di riferimento che descrive la superficie della Terra. Durante il corso dei secoli, con lo sviluppo di nuovi metodi scientifici per misurare la Terra, sono stati creati molti sistemi di riferimento simili a questo. Sono state utilizzate misurazioni terrestri e satellitari per la creazione dei datum, che a loro volta sono state utilizzate per creare proiezioni piane.

I datum geodetici si basano su un'approssimazione della forma della Terra ottenuta dalla rotazione di un ellissoide (definito anche *sferoide*). Uno sferoide è la forma tridimensionale descritta da un'ellissi quando viene ruotata su uno degli assi. Per ulteriori informazioni sugli sferoidi, consultare la sezione "Sistema di coordinate geografiche" a pagina 59.

Tutti gli oggetti spaziali definiti devono essere indicati come riferimento da una determinata colonna. I datum vengono specificati mediante il relativo SRID (spatial reference system identifier). E' possibile scegliere qualsiasi datum supportato da DB2® Geodetic Extender. Lo SRID di questi sistemi è compreso nell'intervallo 2000000000 - 2000001000.

- “Datum supportati da DB2 Geodetic Extender” a pagina 211 contengono i 318 sistemi di riferimento spaziali geodetici forniti da Geodetic Extender.
- E' anche possibile definire un nuovo datum creando un sistema di riferimento spaziale avente un ID compreso nell'intervallo 2000000318 - 2000001000. Per ulteriori informazioni, consultare la sezione “Creazione di un sistema di riferimento spaziale” a pagina 75.

Limitazioni: le funzioni che utilizzano più di un oggetto geo-spaziale come argomenti non possono gestire combinazioni di dati. Geodetic Extender non esegue conversioni di dati.

Argomenti correlati:

- “Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender” a pagina 158
- “Regioni geodetiche” a pagina 161
- “Latitudine e longitudine geodetiche” a pagina 159
- “Distanze geodesiche” a pagina 160
- “Sferoidi geodetici” a pagina 220

Attività correlate:

- “Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo” a pagina 69
- “Creazione di un sistema di riferimento spaziale” a pagina 75

Riferimenti correlati:

- “Datum supportati da DB2 Geodetic Extender” a pagina 211

Latitudine e longitudine geodetiche

Il sistema di riferimento di coordinate utilizzato da DB2 Geodetic Extender, per descrivere le posizioni relative alla Terra, utilizza *latitudine* e *longitudine geodetiche*. La latitudine e la longitudine geodetiche si basano sempre su un datum specifico.

Latitudine geodetica

La latitudine geodetica di un punto è l'angolo tra il piano equatoriale e la linea perpendicolare che interseca la linea normale in un punto sulla superficie della Terra.

Longitudine geodetica

La longitudine geodetica è l'angolo nel piano equatoriale tra la linea *a* che collega il centro della Terra al primo meridiano e la linea *b* che collega il centro al meridiano su cui si trova il punto. Un *meridiano* è una linea diretta sulla superficie del datum che rappresenta la distanza più breve tra i poli.

L'ellissoide nella Figura 17 a pagina 160 mostra gli angoli che rappresentano la latitudine e la longitudine geodetiche. L'angolo per la latitudine geodetica non comincia esattamente al centro a causa della forma ellissoidale della Terra.

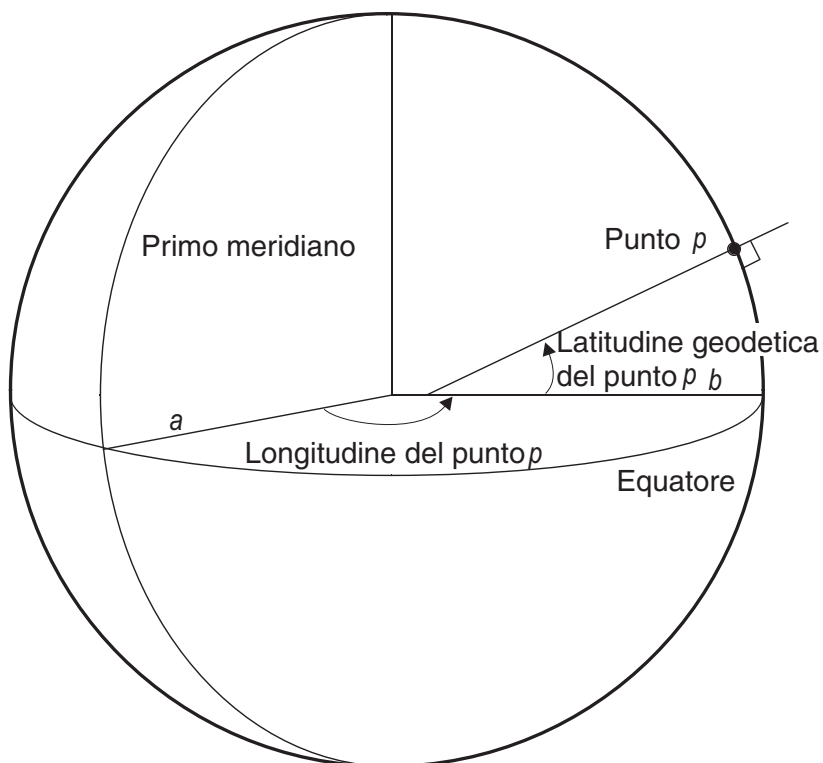


Figura 17. Angoli di latitudine e longitudine geodetiche

Le coordinate della latitudine e della longitudine sono espresse in dati con una frazione decimale. Esistono 360 gradi di longitudine, partendo dal primo meridiano (0° di longitudine) continuando verso est, in direzione positiva, per 180° e ad ovest con valori negativi per -180° . I gradi di latitudine cominciano all'equatore (0° di latitudine) e procedono verso il Polo Nord (90° di latitudine) e il Polo Sud (-90° di latitudine).

Argomenti correlati:

- "DB2 Geodetic Extender" a pagina 157
- "Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender" a pagina 158
- "Regioni geodetiche" a pagina 161
- "Datum geodetici" a pagina 158
- "Distanze geodesiche" a pagina 160
- "Sferoidi geodetici" a pagina 220

Distanze geodesiche

DB2[®] Geodetic Extender misura la distanza tra due punti lungo un *geodesico*. Un geodesico è il percorso più breve tra due punti sulla forma ellissoidale della Terra, che potrebbe non seguire una linea di latitudine costante anche se i due punti finali si trovano alla stessa latitudine.

Poiché i segmenti di linee sono elaborati come geodesici, un poligono di quattro punti molto separati tra loro, come mostrato nella Figura 18 a pagina 161, potrebbero non racchiudere la regione desiderata. Questo poligono copre una regione con linee di longitudine separate da circa 120 gradi, mentre i due punti superiori e inferiori hanno i stessi valori di latitudine. Il valore geodesico tra le due

linee di longitudine segue la curva sulla forma ellissoidale della Terra. La latitudine aumenta insieme al geodesico fino a 20 gradi in più al centro rispetto alle due estremità del geodesico.

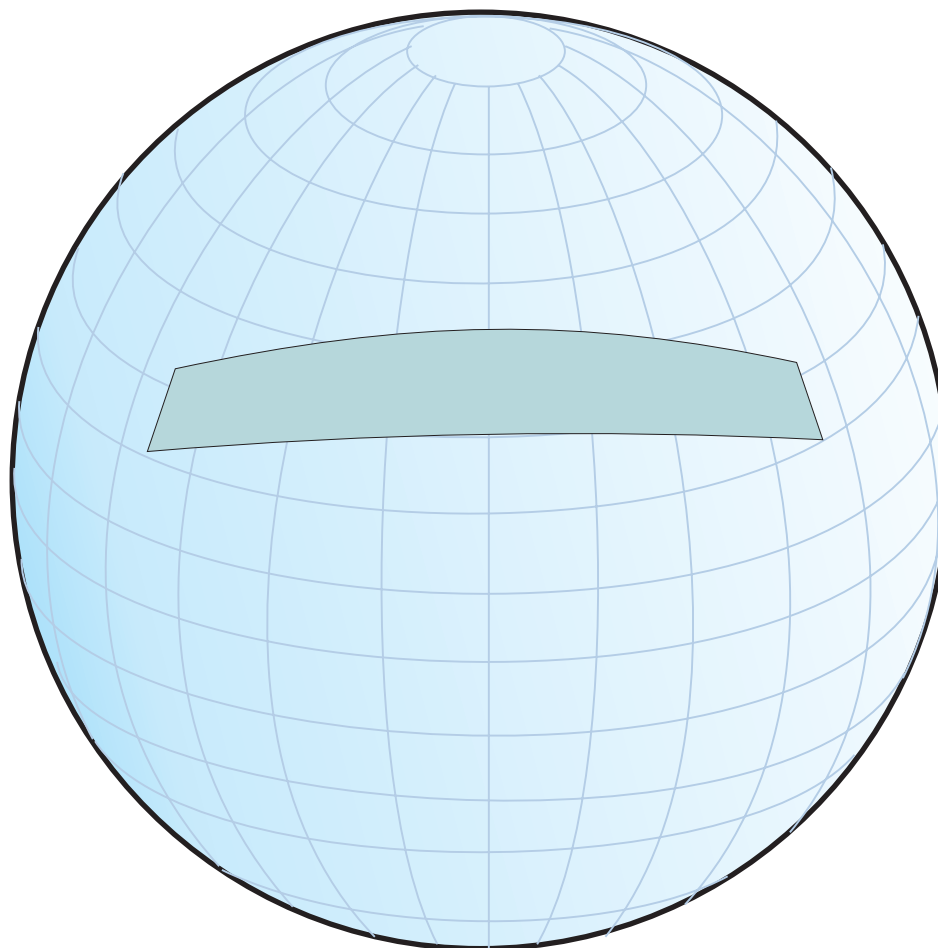


Figura 18. Regione racchiusa da un poligono con punti molto separati

Per rappresentare un percorso non geodesico, ad esempio, se si desidera un segmento di linea che segua una latitudine costante, sarà necessario inserire ulteriori punti intermedi.

Argomenti correlati:

- “Sistema di coordinate geografiche” a pagina 59

Riferimenti correlati:

- “Differenze nell’uso di rappresentazioni della Terra piana o sferica” a pagina 197
- “ST_Distance” a pagina 370

Regioni geodetiche

Una regione geodetica (poligono) è un’area sulla superficie della Terra che ha determinate caratteristiche specifiche per un’applicazione. Esempi di tali regioni possono essere un’area di influenza commerciale o un’area vista da un satellite in un determinato momento.

Geodetic Extender definisce una regione mediante una sequenza ordinata di punti che formano una circonferenza chiusa. L'ordine in cui i punti vengono specificati nel poligono, è decisivo. Seguendo un poligono da vertice a vertice nell'ordine definito, l'area a sinistra è interna al poligono.

E' possibile utilizzare il tipo di dati ST_Polygon per definire una regione racchiusa tra una o più circonferenze, come mostrato dalla Figura 19 a pagina 162. Definire il poligono utilizzando le coordinate di latitudine e longitudine dei punti (vertici) che determinano le circonferenze.

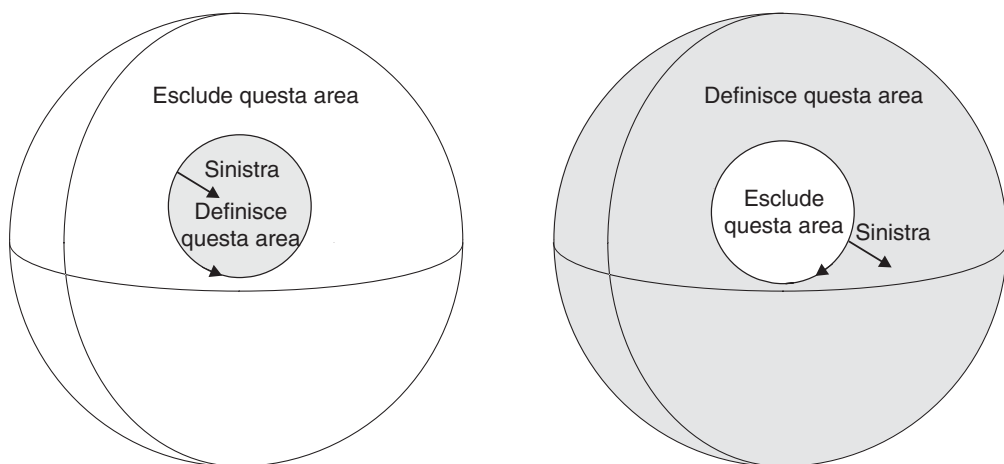


Figura 19. Definizione ed esclusione di aree

Una circonferenza divide la superficie della Terra in due regioni: una interna al poligono e l'altra esterna. Il lato sinistro della Figura 19 mostra una circonferenza con vertici specificati in senso antiorario in modo che tutti i punti a sinistra si trovino all'interno della circonferenza. Il lato destro della figura mostra una circonferenza con vertici in senso orario, in modo che tutti i punti a sinistra si trovino all'esterno della circonferenza.

Per definire una regione come un poligono, sarà necessario specificare l'ordine dei vertici di ciascuna circonferenza in modo che la parte interna del poligono si trovi a sinistra quando si attraversa la circonferenza. Per definire una regione esclusa, specificare i vertici della circonferenza nell'ordine opposto come mostrato nella Figura 20 a pagina 163. La parte interna del poligono si trova sempre a sinistra. La Figura 20 a pagina 163 mostra due circonferenze, una interna all'altra. La circonferenza più grande definisce il limite esterno del poligono ed è tracciata in senso antiorario. La circonferenza più piccola definisce il limite interno ed è tracciata in senso orario.

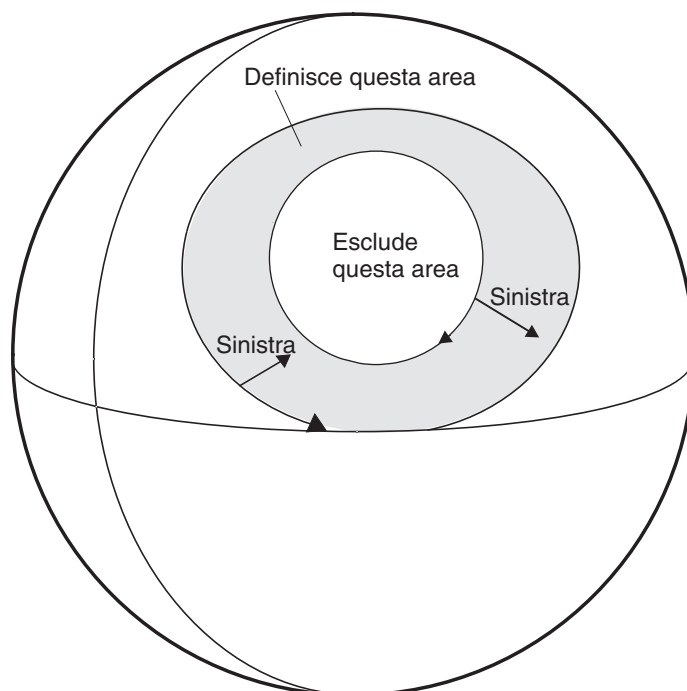


Figura 20. Definizione di un'area con più circonferenze

Se viene creato un poligono più grande dell'emisfero, verrà visualizzato un messaggio di avviso simile a quello riportato di seguito. E' possibile specificare queste dimensioni per il poligono; il messaggio viene visualizzato nel caso in cui sia stato specificato inavvertitamente l'ordine dei vertici non corretto e venga visualizzato un poligono grande invece di uno piccolo.

GSE3733W "Il poligono è più grande della metà della Terra. Verificare l'orientamento orario dei punti dei vertici.

Argomenti correlati:

- "Sistemi di coordinate proiettate" a pagina 65
- "Datum geodetici" a pagina 158
- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- "Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo" a pagina 69
- "Creazione di un sistema di riferimento spaziale" a pagina 75

Capitolo 17. Impostazione di DB2 Geodetic Extender

Questo capitolo contiene istruzioni per l'impostazione di DB2 Geodetic Extender, la migrazione da Informix Geodetic DataBlade e l'inserimento dei dati geodetici nelle colonne spaziali.

Impostazione ed abilitazione di DB2 Geodetic Extender

DB2 Geodetic Extender considera la Terra come un globo, mentre Spatial Extender considera la superficie curva della Terra come una carta geografica piana. Installando Geodetic Extender, sarà possibile analizzare i dati spaziali con più precisione.

Un sistema DB2 Geodetic Extender consiste in DB2 Universal Database, DB2 Spatial Extender, DB2 Geodetic Extender e, per la maggior parte delle applicazioni, un geobrowser.

Suggerimento: per ulteriori informazioni sull'abilitazione di DB2 Geodetic Extender, consultare le *DB2 Release Notes*.

Prerequisiti:

Prima di abilitare DB2 Geodetic Extender, è necessario:

- Installare e configurare DB2 Universal Database™ Enterprise Server Edition Versione 8.2.

E' necessario installare DB2 UDB sul sistema *prima* di installare DB2 Spatial Extender e DB2 Geodetic Extender. Se si desidera utilizzare il Centro di controllo DB2, creare e configurare DB2 Administration Server (DAS). Per ulteriori informazioni sulla creazione e configurazione di DAS, consultare la pubblicazione IBM® *DB2 Universal Database Administration Guide: Implementation*

- Installare e configurare DB2 Spatial Extender.

DB2 Geodetic Extender è integrato nello stesso codice di librerie di DB2 Spatial Extender. Il CD di installazione di Spatial Extender, quindi, include Geodetic Extender. I requisiti di spazio su disco per Spatial Extender comprendono l'installazione di Geodetic Extender. Tuttavia, non sarà possibile utilizzare Geodetic Extender fino a quando non viene acquistata una licenza. Per ulteriori informazioni, consultare le sezioni "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26 e "Impostazione e installazione di Spatial Extender" a pagina 25.

- Se si dispone di un database DB2 Spatial Extender Versione 8.1 database, per utilizzare DB2 Geodetic Extender, sarà necessario migrarlo alla Versione 8.2.

Geodetic Extender ridefinisce diverse funzioni spaziali e definisce ulteriori sistemi di riferimento spaziali per la gestione dei dati geodetici. Il programma di utilità per la migrazione **migrate_v82** consente ad un database esistente abilitato per i dati spaziali, di gestire i dati geodetici. Per ulteriori informazioni, consultare la sezione "Migrazione di un database abilitato per le funzioni spaziali" a pagina 45.

- Acquistare una licenza per DB2 Geodetic Extender.

Impostazione di DB2 Geodetic Extender

Acquistando una licenza per DB2 Geodetic Extender, sarà possibile abilitare la chiave di licenza. Se si desidera acquistare DB2 Geodetic Extender, rivolgersi al rappresentante vendite.

Limitazioni:

DB2 Geodetic Extender viene fornito su licenza solo con DB2 Universal Database™ Enterprise Server Edition Versione 8.2.

Procedura:

Abilitare la licenza per DB2 Geodetic Extender in uno dei seguenti modi:

- Utilizzare il Centro licenze nel Centro di controllo DB2. Visualizzare la guida in linea nel Centro licenze DB2 per ulteriori informazioni sull'abilitazione della licenza di Geodetic Extender.
- Eseguire il comando **db2licm**.

Dopo aver abilitato la licenza per DB2 Geodetic Extender, "Inserimento dei dati geodetici nelle colonne spaziali" a pagina 173.

Argomenti correlati:

- "Requisiti di sistema per l'installazione di Spatial Extender" a pagina 26

Attività correlate:

- "Impostazione e installazione di Spatial Extender" a pagina 25
- "Migrazione di un database abilitato per le funzioni spaziali" a pagina 45
- "Inserimento dei dati geodetici nelle colonne spaziali" a pagina 173

Riferimenti correlati:

- "CD per dati e carte geografiche di DB2 Spatial Extender" a pagina 42

Migrazione da Informix Geodetic DataBlade a DB2 Geodetic Extender

Se si utilizza IBM Informix Geodetic DataBlade per memorizzare e manipolare gli oggetti geospaziali in un database, è possibile migrare i dati e le applicazioni a IBM DB2 Geodetic Extender, ma vi saranno alcune limitazioni.

Prerequisiti:

E' necessario che le applicazioni Geodetic DataBlade possano utilizzare i tipi di dati e le funzioni di DB2 Geodetic Extender.

Limitazioni:

Se si utilizza Informix Geodetic DataBlade, è possibile migrare i dati a DB2 Geodetic Extender se si soddisfano i seguenti criteri:

- Utilizzare solo tipi di dati GeoPoint, GeoLineseg, GeoString, GeoRing e GeoPolygon.
- Utilizzare solo le funzioni Geodetic DataBlade che hanno controparti equivalenti o simili in DB2 Geodetic Extender, come le tabelle descritte di seguito.
- Indicizzare solo il componente spaziale di GeoObjects; in altre parole, non indicizzare intervalli di ore o altitudine.

Procedura:

Per migrare da IBM Informix Geodetic DataBlade a IBM DB2 Geodetic Extender:

1. Riscrivere le istruzioni SQL affinché utilizzino i tipi di dati e le funzioni di DB2 Geodetic Extender. Per i tipi di dati e funzioni corrispondenti, fare riferimento alla seguente tabella:
 - Tabella 14
 - Tabella 15 a pagina 168
 - Tabella 16 a pagina 168
 - Tabella 17 a pagina 169
 - Tabella 18 a pagina 170
 - Tabella 19 a pagina 170
 - Tabella 20 a pagina 170
 - Tabella 21 a pagina 170
 - Tabella 22 a pagina 171
 - Tabella 23 a pagina 171
2. Caricare o importare i dati in DB2 Geodetic Extender.
3. Riscrivere le applicazioni che utilizzano Informix ODBC, ESQL/C e JDBC. La Tabella 24 a pagina 172 mostra la connettività del client corrispondente in Geodetic DataBlade e Geodetic Extender.

Tabella 14. Tipi di dati corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Tipo di dati in Informix Geodetic DataBlade	Tipo di dati corrispondente in DB2 Geodetic Extender	Commenti per i tipi di dati simili
GeoBox		Convertire prima in un a GeoPolygon in Geodetic DataBlade, quindi utilizzare ST_Polygon in Geodetic Extender
GeoCircle		Convertire prima in un GeoPolygon, quindi migrare a ST_Polygon
GeoEllipse		Convertire prima in un GeoPolygon, quindi migrare a ST_Polygon
GeoLineseg	ST_LineString	
GeoObject	ST_Geometry	ST_Geometry e i relativi tipi secondari non supportano i tipi di dati GeoAltRange e GeoTimeRange.
GeoPoint	ST_Point	
GeoPolygon	ST_MultiPolygon, ST_Polygon	ST_MultiPolygon richiede un punto di chiusura esplicito per ciascuna circonferenza. Se un GeoPolygon ha una circonferenza esterna, può essere associato a ST_Polygon.
GeoRing	ST_LineString	
GeoString	ST_LineString	

I seguenti tipi di dati di Geodetic DataBlade non hanno un tipo di dati corrispondente in Geodetic Extender:

- GeoAltitude
- GeoAltRange
- GeoAngle

Impostazione di DB2 Geodetic Extender

- GeoAzimuth
- GeoBox
- GeoCircle
- GeoCoords
- GeoDistance
- GeoEllipse
- GeoLatitude
- GeoLongitude
- GeoTimeRange
- GeoVoronoi

Tabella 15. Funzioni predicato corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
Contains	ST_Contains
Inside	ST_Within
Intersect	ST_Intersects
Outside	ST_Disjoint
Within	ST_Distance

Le seguenti funzioni di Geodetic DataBlade non hanno una funzione corrispondente in Geodetic Extender:

- Beyond
- Equal
- Nearest

Tabella 16. Funzioni di produzione corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender	Commenti per le funzioni simili
Difference	ST_Difference	ST_Difference supporta punti in aggiunta a poligoni
Generalize	ST_Generalize	
Intersection	ST_Intersection	ST_Intersection(line,line) potrebbe ottenere un multipunto come risultato. ST_Intersection (line,poly) potrebbe ottenere una multilinea come risultato. Restituisce Vuoto per gli oggetti non uniti.
SymDifference	ST_SymDifference	ST_SymDifference supporta punti in aggiunta a poligoni
Union	ST_Union	ST_Union supporta punti e linee in aggiunta a poligoni

Tabella 17. Funzioni di accesso corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender	Commenti per le funzioni simili
Center	ST_MidPoint, ST_PointOnSurface	ST_MidPoint è un valido sostituto per le linee. ST_PointOnSurface è un valido sostituto per i poligoni.
Coords	ST_PointN	
La dimensione	ST_Dimension	
HasZValue	ST_Is3d	
IsGeoBox	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoCircle	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoEllipse	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoLineseg	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoPoint	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoPolygon	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoRing	Utilizzare l'espressione IS OF o ST_GeometryType	
IsGeoString	Utilizzare l'espressione IS OF o ST_GeometryType	
Latitude	ST_Y	
Longitude	ST_X	
NPoints	ST_NumPoints	
NRings	ST_NumGeometries, ST_NumInteriorRing	Utilizzare ST_NumGeometries per ottenere il numero totale delle circonferenze esterne e sommare ST_NumInteriorRings per ciascun poligono nell'insieme di multipoligoni
Ring	ST_GeometryN, ST_ExteriorRing, ST_InteriorRingN	Utilizzare ST_GeometryN insieme a ST_ExteriorRing e ST_InteriorRingN
SRID	ST_SRID	
Zvalue	ST_Z	

Le seguenti funzioni di Geodetic DataBlade non hanno funzioni di accesso corrispondenti in Geodetic Extender:

- IsLarge
- IsSmallArea

Impostazione di DB2 Geodetic Extender

Tabella 18. Funzioni di modifica corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
SetSRID	ST_SRID

Le seguenti funzioni di Geodetic DataBlade non hanno funzioni di modifica corrispondenti in Geodetic Extender:

- SetAltRange
- SetAltRangeZ
- SetDist
- SetTimeRange

Tabella 19. Funzioni di misura corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
Area	ST_Area
Distance	ST_Distance
Length	ST_Length, ST_Perimeter

La seguente funzione di misura VoronoiResolution non ha una funzione corrispondente in Geodetic Extender.

Tabella 20. Funzioni di diminuzione corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
GeoBox	Utilizzare l'espressione SQL TREAT
GeoCircle	Utilizzare l'espressione SQL TREAT
GeoEllipse	Utilizzare l'espressione SQL TREAT
GeoLineseg	Utilizzare l'espressione SQL TREAT
GeoPoint	Utilizzare l'espressione SQL TREAT
GeoPolygon	Utilizzare l'espressione SQL TREAT
GeoRing	Utilizzare l'espressione SQL TREAT
GeoString	Utilizzare l'espressione SQL TREAT

Tabella 21. Funzioni di costruzione corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
GeoCoords	ST_Point
GeoPoint	ST_Point

Le seguenti funzioni di Geodetic DataBlade non hanno funzioni di costruzione corrispondenti in Geodetic Extender:

- GeoBox

- GeoCircle
- GeoEllipse
- GeoLineSeg

Tabella 22. Funzioni di diagnostica corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Funzioni in Informix Geodetic DataBlade	Funzioni corrispondenti in DB2 Geodetic Extender
GeoTraceLevel	Funzione di traccia DB2
IsValidGeometry	ST_IsValid

Le seguenti funzioni di diagnostica Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- GeoInRowSize
- GeoOutOfRowSize
- GeoRelease
- GeoTotalSize
- GeoTraceLevelSet
- GeoWarningLevel
- GeoWarningLevelSet
- IsValidSDTS

Tabella 23. Tabelle del catalogo di sistema corrispondenti in Informix Geodetic DataBlade e Geodetic Extender

Tabella del catalogo di sistema in Informix Geodetic DataBlade	Vista del catalogo corrispondente in DB2 Geodetic Extender
GeoLenUnit	DB2GSE.ST_UNITS_OF_MEASURE
GeoSpatialRef	DB2GSE.SPATIAL_REF_SYS

Le seguenti tabelle del catalogo di sistema di Geodetic DataBlade non hanno una tabella o vista corrispondente in Geodetic Extender:

- GeoEllipsoid
- GeoParam
- GeoVoronoi

Le seguenti funzioni parametro impostabile dall'utente di Geodetic DataBlade non hanno una funzione corrispondente in Geodetic Extender:

- GeoParamSessionGet
- GeoParamSessionSet

Le seguenti funzioni AltRange di Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- AltRange
- Bottom
- Contains
- Equal
- Inside
- Intersect

Impostazione di DB2 Geodetic Extender

- IsAny
- Outside
- Top

Le seguenti funzioni di TimeRange di Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- Begin
- Contains
- End
- Equal
- IsAny
- Inside
- Intersect
- Outside
- TimeRange

Le seguenti funzioni di ellisse di Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- Azimuth
- Coords
- Major
- Minor

Le seguenti funzioni di circonferenza di Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- Coords
- Radius

Le seguenti funzioni di angolo aritmetico di Geodetic DataBlade non hanno funzioni corrispondenti in Geodetic Extender:

- Divide
- Minus
- Negate
- Plus
- Times

Tabella 24. Prodotti di connettività client corrispondenti in Geodetic DataBlade e DB2 Geodetic Extender

Connettività client in Informix Geodetic DataBlade	Connettività client corrispondente in DB2 Geodetic Extender
ESQLC	SQC
ODBC	ODBC
JDBC	JDBC

La seguente connettività client di Geodetic DataBlade non ha una connettività client corrispondente in Geodetic Extender:

- Java API
- LIBMI

Inserimento dei dati geodetici nelle colonne spaziali

Dopo aver creato le colonne spaziali e aver registrato le colonne per le quali si desidera creare un indice spaziale, sarà possibile inserire nelle colonne i dati geodetici. E' possibile fornire i dati geodetici come segue:

- Importare i seguenti formati di dati in una tabella nuova o esistente:
 - Shape
 - SDE
- Inserire o aggiornare i valori nei seguenti formati di dati:
 - Shape
 - SDE
 - Well-known text (WKT)
 - Well-known binary (WKB)
 - GML (Geography Markup Language)

Limitazioni:

- Per Spatial Extender Versione 8.2, non è possibile utilizzare i comando del geocoder o le procedure memorizzate per trasformare i dati in dati geodetici.
- Per il comportamento geodetico, utilizzare sistemi di riferimento spaziali i cui SRID siano compresi nell'intervallo 2.000.000.000 - 2.000.001.000. Per ulteriori informazioni, consultare la sezione "Sistemi di riferimento spaziali" a pagina 68.
- I dati shape e i dati di trasferimento SDE devono trovarsi in un sistema di coordinate geografiche. Per ulteriori informazioni, consultare la sezione "Sistema di coordinate geografiche" a pagina 59.

Procedura:

La procedura di importazione per i dati geodetici è la stessa utilizzata per i dati spaziali. Per i dettagli, consultare la sezione "Importazione dei dati shape in una tabella nuova o esistente" a pagina 88 e "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90.

Argomenti correlati:

- "Sistemi di riferimento spaziali" a pagina 68
- "Sistema di coordinate geografiche" a pagina 59

Attività correlate:

- "Importazione dei dati shape in una tabella nuova o esistente" a pagina 88
- "Importazione dei dati di trasferimento SDE in una nuova tabella o in una tabella esistente" a pagina 90
- "Registrazione delle colonne spaziali" a pagina 85

Impostazione di DB2 Geodetic Extender

Capitolo 18. Indici geodetici

E' possibile creare indici geodetici Voronoi che consentono di migliorare le prestazioni delle query eseguite nei dati geodetici. Il presente capitolo:

- Descrive gli indici geodetici Voronoi
- Descrive le strutture delle celle Voronoi e indica quando è possibile selezionare una struttura alternativa.
- Spiega come creare un indice geodetico Voronoi.

Indici geodetici Voronoi

DB2[®] Geodetic Extender fornisce un indice geodetico Voronoi che velocizza l'accesso ai dati geodetici. Tale indice organizza l'accesso ai dati geodetici utilizzando la tassellazione Voronoi della superficie terrestre. Per ulteriori informazioni, consultare la sezione "Strutture delle celle Voronoi" a pagina 176.

Geodetic Extender calcola l'MBC (minimum bounding circle) di ciascuna forma geometrica. L'MBC è un cerchio che circonda una forma geometrica geodetica. L'indice Voronoi utilizza queste informazioni fornite dall'MBC per organizzare i dati in una struttura di celle. Una ricerca eseguita utilizzando un indice Voronoi, può arrivare velocemente nei dati organizzati e individuare oggetti nell'area di interesse generale, quindi eseguire verifiche più specifiche sugli oggetti stessi. Un indice Voronoi migliora le prestazioni perché non ha bisogno di esaminare gli oggetti esterni all'area di interesse. Se l'interrogazione viene eseguita senza utilizzare un indice Voronoi, l'interrogazione dovrà valutare tutti gli oggetti per individuare quelli che corrispondono ai criteri specificati.

L'ottimizzatore utilizzerà l'indice geodetico Voronoi con tutte le interrogazioni che contengono le seguenti funzioni nella clausola WHERE:

- EnvelopesIntersect
- ST_Contains
- ST_Distance
- ST_EnvIntersects
- ST_Intersects
- ST_MBRIntersects
- ST_Within

Per ulteriori informazioni, consultare la sezione "Funzioni che utilizzano indici per l'ottimizzazione delle interrogazioni" a pagina 122.

Durante la creazione degli indici geodetici Voronoi, è possibile scegliere una struttura di celle Voronoi alternativa. Per informazioni, consultare la sezione "Considerazioni per la selezione di una struttura di celle Voronoi alternativa" a pagina 177.

Argomenti correlati:

- "Strutture delle celle Voronoi" a pagina 176
- "Considerazioni per la selezione di una struttura di celle Voronoi alternativa" a pagina 177

Indici geodetici

- “Indici di griglia spaziali” a pagina 100

Attività correlate:

- “Creazione di indici geodetici Voronoi” a pagina 179

Riferimenti correlati:

- “Strutture delle celle Voronoi fornite con DB2 Geodetic Extender” a pagina 182
- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

Strutture delle celle Voronoi

Per eseguire le elaborazioni in modo efficace, DB2® Geodetic Extender divide la superficie della Terra in celle più piccole, esagonali, più pratiche. La suddivisione è nota come *tassellazione Voronoi*, e la struttura di dati che descrive tale suddivisione è chiamata *struttura di celle Voronoi*. Una *tassellazione Voronoi* è una struttura di celle in cui la parte interna di ciascuna cella è formata da tutti i punti più vicini a un determinato punto del reticolato rispetto a qualsiasi altro punto. Le celle di una struttura di celle Voronoi sono *poligoni convessi*. Un poligono convesso di una serie di punti è l’insieme convesso più piccolo che racchiude i punti (o il poligono più piccolo che definisce la parte “esterna” di un gruppo di punti). Le strutture di celle Voronoi tendono a essere poligoni di forme irregolari; il numero e la posizione delle celle vengono modificate per adattarsi alla densità e alla posizione dei dati spaziali.

Ad esempio, una struttura di celle Voronoi, può suddividere la Terra in poligoni basati sulla popolazione mondiale. Quando la popolazione (e i dati) è più densa, vengono creati poligoni più piccoli. Quando la popolazione è meno fitta, vengono creati poligoni più grandi.

La Figura 21 a pagina 177 mostra la struttura Voronoi basata sulla densità della popolazione mondiale. Geodetic Extender utilizza questa struttura di celle per le elaborazioni spaziali.

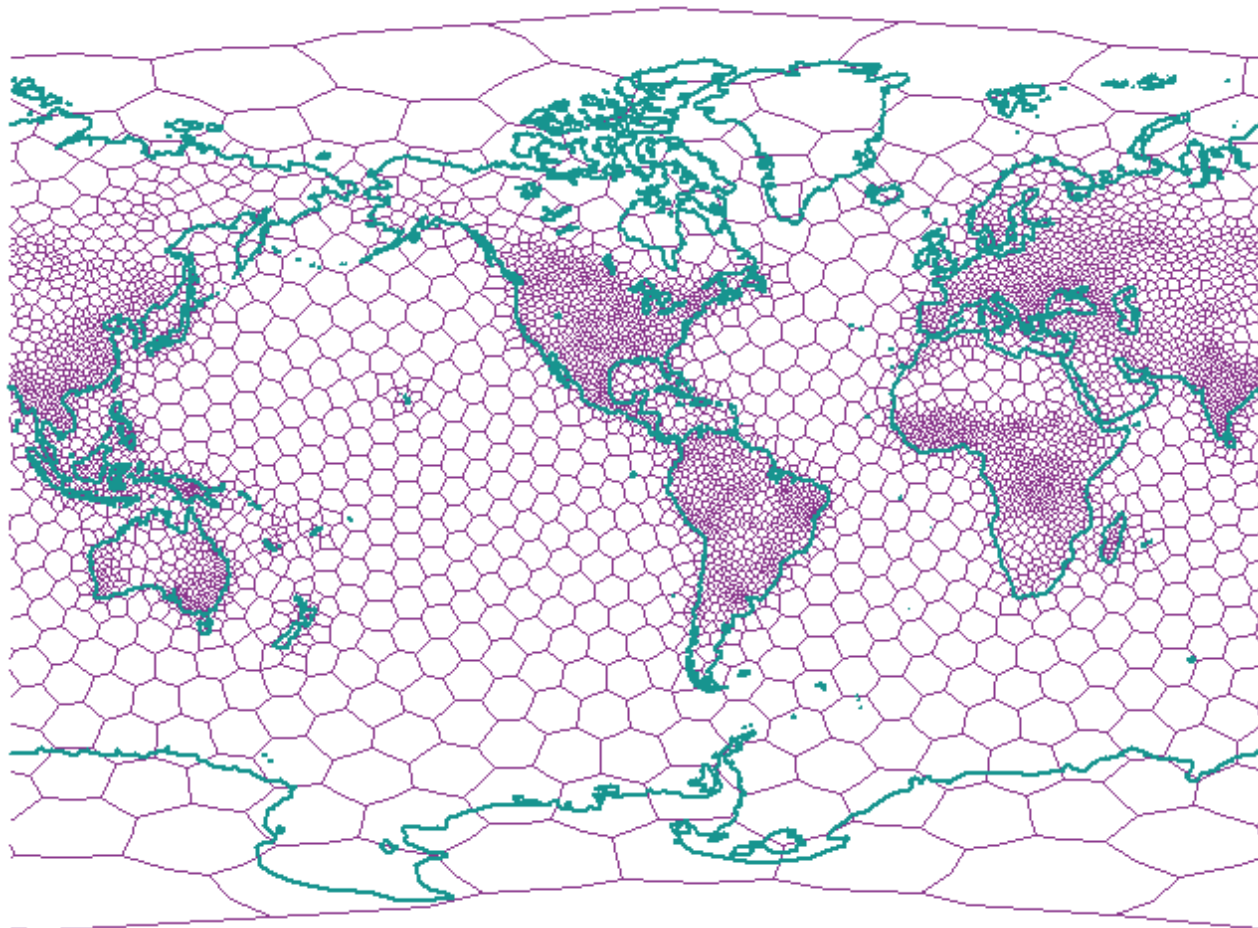


Figura 21. Struttura Voronoi basata sulla densità della popolazione mondiale

Argomenti correlati:

- “Indici geodetici Voronoi” a pagina 175
- “Considerazioni per la selezione di una struttura di celle Voronoi alternativa” a pagina 177

Attività correlate:

- “Creazione di indici geodetici Voronoi” a pagina 179

Riferimenti correlati:

- “Istruzione CREATE INDEX per l’indice geodetico Voronoi” a pagina 180
- “Strutture delle celle Voronoi fornite con DB2 Geodetic Extender” a pagina 182

Considerazioni per la selezione di una struttura di celle Voronoi alternativa

Tutte le operazioni eseguite nelle forme geometriche geodetiche utilizzano l’ID Voronoi 1 che indica le strutture di celle Voronoi basata sulla densità della popolazione mondiale. Quando viene creato un indice, se i dati sono raggruppati in una o più aree della Terra, ad esempio dati relativi alle strade di uno o più paesi, sarà possibile scegliere una struttura di celle Voronoi alternativa, contenente celle più piccole nelle aree in cui si trovano i dati (perché la risoluzione è

inversamente proporzionale alla dimensione della cella). DB2[®] Geodetic Extender fornisce diverse strutture di celle Voronoi per l'indicizzazione che possono meglio adattarsi ai dati. Per un elenco delle strutture alternative e diagrammi disponibili, che illustrano tali strutture di celle, consultare la sezione "Strutture delle celle Voronoi fornite con DB2 Geodetic Extender" a pagina 182.

Limitazione: è possibile scegliere una struttura di celle Voronoi alternativa solo quando viene creato un indice Voronoi geodetico.

La struttura dodeca04 (ID Voronoi 12) è quella che meglio si adatta ai dati distribuiti uniformemente sull'intera superficie terrestre, ad esempio un'immagine satellitare. Le dimensioni delle celle sono grossolanamente uniformi e la risoluzione peggiore è di circa 10 centimetri. Se i dati dell'applicazione corrispondono a una delle seguenti condizioni, utilizzare una struttura di celle Voronoi diversa dalla struttura della popolazione mondiale predefinita (ID Voronoi 1) o dalla struttura dodeca04:

Alta risoluzione

Se si desidera determinare l'eventuale intersezione di oggetti distanti tra loro meno di 10 centimetri, sarà necessario utilizzare una struttura di celle Voronoi che ha celle più piccole nella regione in cui si trovano i dati. La risoluzione è inversamente proporzionale alla dimensione della cella.

Poligono con molti vertici

Se i dati consistono in poligoni con un alto numero di vertici e si trovano in un'area relativamente piccola, è possibile passare ad una struttura di celle Voronoi che abbia più celle nella regione di interesse. Se la maggior parte dei poligoni hanno meno di 50 vertici, non sarà necessario cambiare struttura. Se gli unici poligoni dell'insieme di dati che hanno molti vertici corrispondono a continenti, anche in questo caso non sarà necessario cambiare struttura.

Se si dispone di poligoni con più di 3000 vertici, aventi dimensione delle contee degli Stati Uniti, è possibile migliorare di molto le prestazioni dell'interrogazione passando a una diversa struttura di celle Voronoi, in particolare se l'applicazione esegue diverse interrogazioni in poligoni che si intersecano.

Dati molto densi

Se i dati sono concentrati in regioni molto piccole (ad esempio, centinaia di oggetti per chilometro quadrato) sarà possibile migliorare le prestazioni dell'interrogazione utilizzando una struttura di celle Voronoi la cui densità di celle corrisponda alla densità dei dati.

Argomenti correlati:

- "Indici geodetici Voronoi" a pagina 175
- "Strutture delle celle Voronoi" a pagina 176

Attività correlate:

- "Creazione di indici geodetici Voronoi" a pagina 179

Riferimenti correlati:

- "Istruzione CREATE INDEX per l'indice geodetico Voronoi" a pagina 180
- "Strutture delle celle Voronoi fornite con DB2 Geodetic Extender" a pagina 182

Creazione di indici geodetici Voronoi

DB2 Geodetic Extender fornisce un nuovo metodo di accesso spaziale che consente di creare indici su colonne contenenti dati geodetici. Le interrogazioni che utilizzano gli indici vengono eseguite più velocemente.

Prerequisiti:

Prima di creare un indice geodetico Voronoi, l'ID utente deve disporre delle stesse autorizzazioni e privilegi necessarie per la creazione di un indice di griglia spaziale (consultare la sezione "Creazione degli indici di griglia spaziali" a pagina 106).

Limitazioni:

Durante la creazione di un indice geodetico Voronoi, valgono le stesse limitazioni previste per la creazione degli indici utilizzando l'istruzione CREATE INDEX, ovvero la colonna in cui viene creato l'indice dovrà essere la colonna di una tabella di base non una colonna di viste o con nickname. DB2 UDB risolverà gli alias durante il processo.

Procedura:

E' possibile creare un indice geodetico Voronoi in uno dei seguenti modi:

- Utilizzando la finestra di creazione indice del Centro di controllo DB2.
- Utilizzando l'istruzione SQL CREATE INDEX con l'estensione db2gse.spatial_index nella clausola EXTEND USING.

Per creare un'indice geodetico Voronoi utilizzando il Centro di controllo, procedere come segue:

1. Nel Centro di controllo, fare clic con il tasto destro del mouse sulla tabella della colonna spaziale per la quale si desidera creare l'indice geodetico Voronoi, e selezionare **Spatial Extender** → **Indici spaziali** dal menu. Viene aperta la finestra Indici spaziali.
2. Attenersi alle istruzioni riportate nella guida in linea relativa alla finestra. Per visualizzare le istruzioni, fare clic sul pulsante ? nella finestra Indici spaziali.

Per creare un'indice geodetico Voronoi utilizzando l'istruzione SQL CREATE INDEX, procedere come segue:

Eseguire l'istruzione CREATE INDEX con la clausola EXTEND USING e l'estensione indice di griglia db2gse.spatial_index.

Esempio:

Il seguente esempio di istruzione CREATE INDEX crea l'indice geodetico STORESX1 nella colonna spaziale LOCATION nella tabella CUSTOMERS:

```
CREATE INDEX storesx1
  ON customers (location)
  EXTEND USING db2gse.spatial_index (-1, -1, 1)
```

Per gli indici geodetici Voronoi, sarà necessario specificare il valore -1 nei primi due parametri della clausola USING db2gse.spatial_index. Per ulteriori informazioni, consultare la sezione "Istruzione CREATE INDEX per l'indice geodetico Voronoi" a pagina 180.

Argomenti correlati:

Indici geodetici

- “Indici geodetici Voronoi” a pagina 175
- “Strutture delle celle Voronoi” a pagina 176
- “Considerazioni per la selezione di una struttura di celle Voronoi alternativa” a pagina 177

Attività correlate:

- “Creazione degli indici di griglia spaziali” a pagina 106

Riferimenti correlati:

- “Istruzione CREATE INDEX per l’indice geodetico Voronoi” a pagina 180
- “Strutture delle celle Voronoi fornite con DB2 Geodetic Extender” a pagina 182
- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

Istruzione CREATE INDEX per l’indice geodetico Voronoi

Utilizzare l’istruzione CREATE con la clausola EXTEND USING per creare un indice geodetico Voronoi.

Sintassi:

```
▶ CREATE INDEX [schema_indice.] nome_indice ON  
[schema_tabella.] nome_tabella (— nome_colonna —) EXTEND USING  
▶ db2gse.spatial_index (— -1 —, — -1 —, — Voronoi_ID —) ▶
```

Dove:

schema_indice

Nome dello schema a cui appartiene l’indice creato. Se non viene specificato alcun nome, DB2 UDB utilizzerà il nome dello schema memorizzato nel registro speciale CURRENT SCHEMA.

nome_indice

Nome semplice dell’indice geodetico che si desidera creare.

schema_tabella

Nome dello schema a cui appartiene la tabella che contiene la *nome_colonna*. Se non viene specificato alcun nome, DB2 UDB utilizzerà il nome dello schema memorizzato nel registro speciale CURRENT SCHEMA.

nome_tabella

Nome della tabella che contiene la *nome_colonna*.

nome_colonna

Nome della colonna spaziale in cui viene creato l’indice Voronoi.

Voronoi_ID

Un numero intero che identifica l’ID della struttura di celle Voronoi. Sono disponibili quattordici strutture di celle Voronoi. L’ID Voronoi 1 indica la struttura di celle Voronoi basata sulla densità della popolazione mondiale, utilizzata anche per tutte le operazioni spaziali da DB2 Geodetic Extender.

Esempi:

Il seguente esempio di istruzione CREATE INDEX crea l'indice geodetico STORESX1 nella colonna spaziale LOCATION nella tabella CUSTOMERS:

```
CREATE INDEX storesx1
  ON customers (location)
  EXTEND USING db2gse.spatial_index (-1, -1, 1)
```

L'ottimizzatore utilizzerà l'indice Voronoi con tutte le interrogazioni che contengono le seguenti funzioni nella clausola WHERE:

- ST_Contains
- ST_Distance
- ST_Intersects
- ST_MBRIntersects
- ST_EnvIntersects
- EnvelopesIntersect
- ST_Within

Le seguenti istruzioni mostrano come utilizzare un indice Voronoi. Innanzitutto, inserire i dati nella tabella CUSTOMER. E' possibile inserire i valori direttamente, come mostrato in questa prima istruzione INSERT:

```
INSERT INTO customer
(id, last_name, first_name, address, city, state, zip,
location)
VALUES
('123-456789', 'Duck', 'Donald',
'123 Mallard Way', 'Wetland Marsh', 'ND', '55555-5555',
db2gse.ST_GeomFromWKT('POINT(123.123, 45.67)', 2000000000))
```

In alternativa, per inserire i valori in una tabella, è possibile utilizzare le variabili in un'applicazione, come mostrato nell'interrogazione successiva:

```
INSERT INTO customer
(id, last_name, first_name,
address, city, state, zip,
location)
VALUES
(:mid, :mlast, :mfirst,
:maddress, :mcity, :mstate, :mzip,
db2gse.ST_GeomFromWKB(:mlocation))
```

La seguente istruzione UPDATE, modifica i dati inseriti. Non utilizza l'indice STORESX1 perché nella clausola WHERE non vengono utilizzate le funzioni ST_Contains, ST_Distance, ST_Intersects, ST_MBRIntersects, ST_EnvIntersects, EnvelopesIntersect o ST_Within.

```
UPDATE customer
SET location = db2gse.ST_GeomFromWKT('POINT(123.123, 45.67)',
2000000000)
WHERE id = '123-456789';
```

Le seguenti istruzioni DELETE possono utilizzare l'indice STORESX1, se l'ottimizzatore determina che l'indice migliora le prestazioni in quanto le istruzioni DELETE utilizzano la funzione ST_Within e le funzioni ST_Intersects nelle proprie clausole WHERE, rispettivamente:

```
DELETE FROM customers
WHERE db2gse.ST_Within(location, :BayArea) = 1;
DELETE FROM customers
WHERE db2gse.ST_Intersects(c.location, :BayArea) = 1
```

Indici geodetici

Anche le due seguenti istruzioni SELECT possono utilizzare l'indice STORESX1:

```
SELECT s.id, AVG(c.location..ST_Distance(s.location))
FROM customers c, stores s
WHERE db2gse.ST_Within(c.location, s.zone) = 1
GROUP BY s.id;
SELECT c.location..ST_AsText()
FROM customers c
WHERE db2gse.ST_Within(c.location, :BayArea) = 1
```

Argomenti correlati:

- "Indici geodetici Voronoi" a pagina 175
- "Strutture delle celle Voronoi" a pagina 176
- "Considerazioni per la selezione di una struttura di celle Voronoi alternativa" a pagina 177

Attività correlate:

- "Creazione di indici geodetici Voronoi" a pagina 179

Riferimenti correlati:

- "Strutture delle celle Voronoi fornite con DB2 Geodetic Extender" a pagina 182

Strutture delle celle Voronoi fornite con DB2 Geodetic Extender

Ciascuna struttura di celle Voronoi copre l'intero pianeta. Nelle illustrazioni seguenti, vengono mostrate solo quelle zone della Terra nell'area in cui le celle sono dense per quella particolare struttura di celle Voronoi. Quando viene selezionata una struttura di celle Voronoi, tenere presente che le celle esterne alle aree illustrate, saranno ampie e avranno una risoluzione bassa. Se i dati si trovano in queste aree date, le prestazioni delle interrogazioni potrebbero peggiorare.

La seguente tabella elenca le strutture di celle Voronoi fornite da DB2 Geodetic Extender. Tali strutture di celle Voronoi sono fornite dalla Geodyssey Ltd.

Tabella 25. Strutture di celle Voronoi

Descrizione	ID Voronoi	Illustrazione
Mondiale, basata sulla densità della popolazione	1	Figura 22 a pagina 183
Stati Uniti	2	Figura 23 a pagina 184
Canada	3	Figura 24 a pagina 185
India	4	Figura 25 a pagina 186
Giappone	5	Figura 26 a pagina 187
Africa	6	Figura 27 a pagina 188
Australia	7	Figura 28 a pagina 189
Europa	8	Figura 29 a pagina 190
America del Nord	9	Figura 30 a pagina 191
America del Sud	10	Figura 31 a pagina 192
Mediterraneo	11	Figura 32 a pagina 193
Mondiale, distribuzione dati uniforme, risoluzione media (dodeca04)	12	Figura 33 a pagina 194

Tabella 25. Strutture di celle Voronoi (Continua)

Descrizione	ID Voronoi	Illustrazione
Mondiale, basata sulle risorse industriali (nazioni G7)	13	Figura 34 a pagina 195
Mondiale, distribuzione dati uniforme, risoluzione bassa (isotipo)	14	Figura 35 a pagina 196

Mondiale, basata sulla densità di popolazione (ID Voronoi: 1)

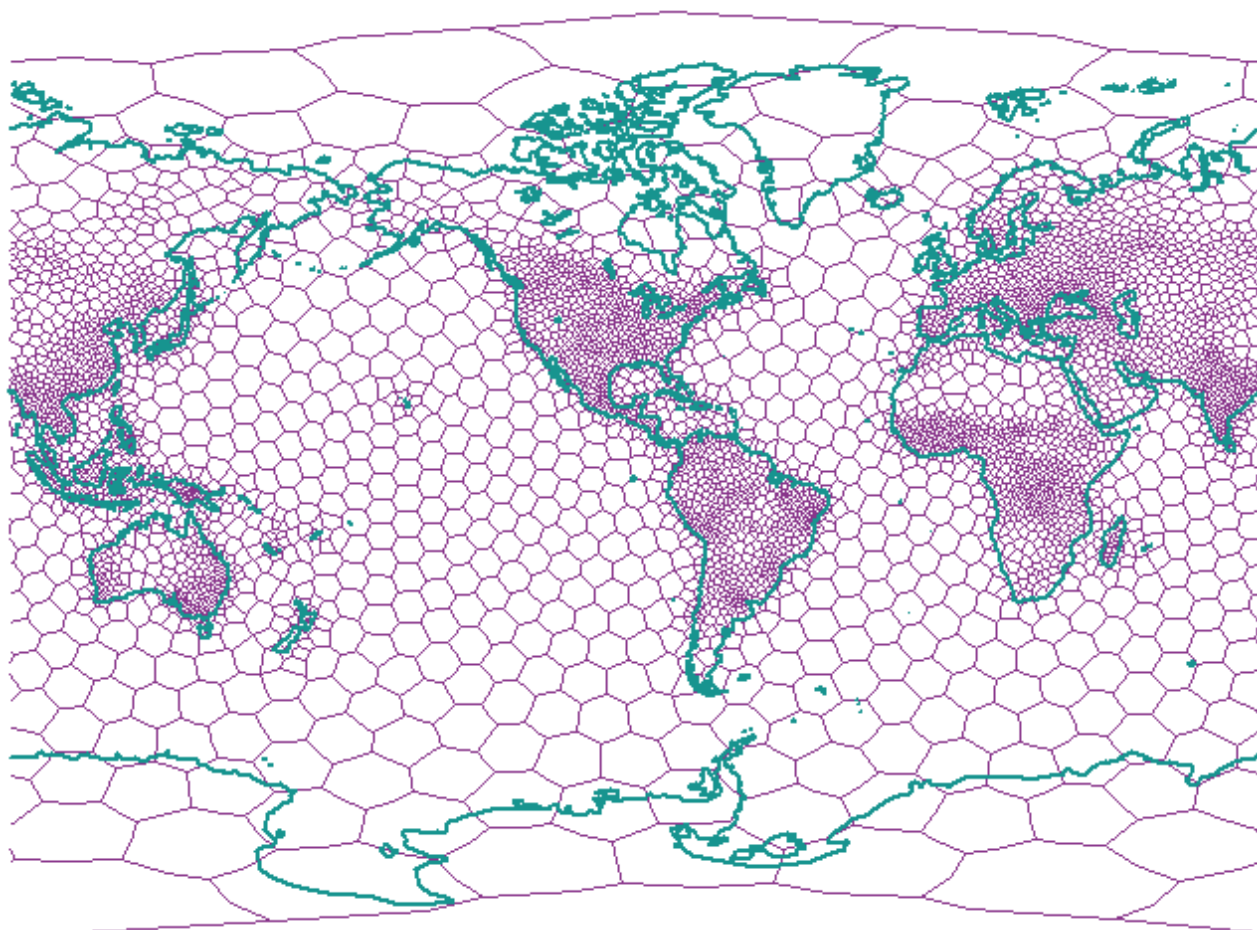


Figura 22. Struttura di celle Voronoi per la popolazione mondiale

Stati Uniti (ID Voronoi: 2)

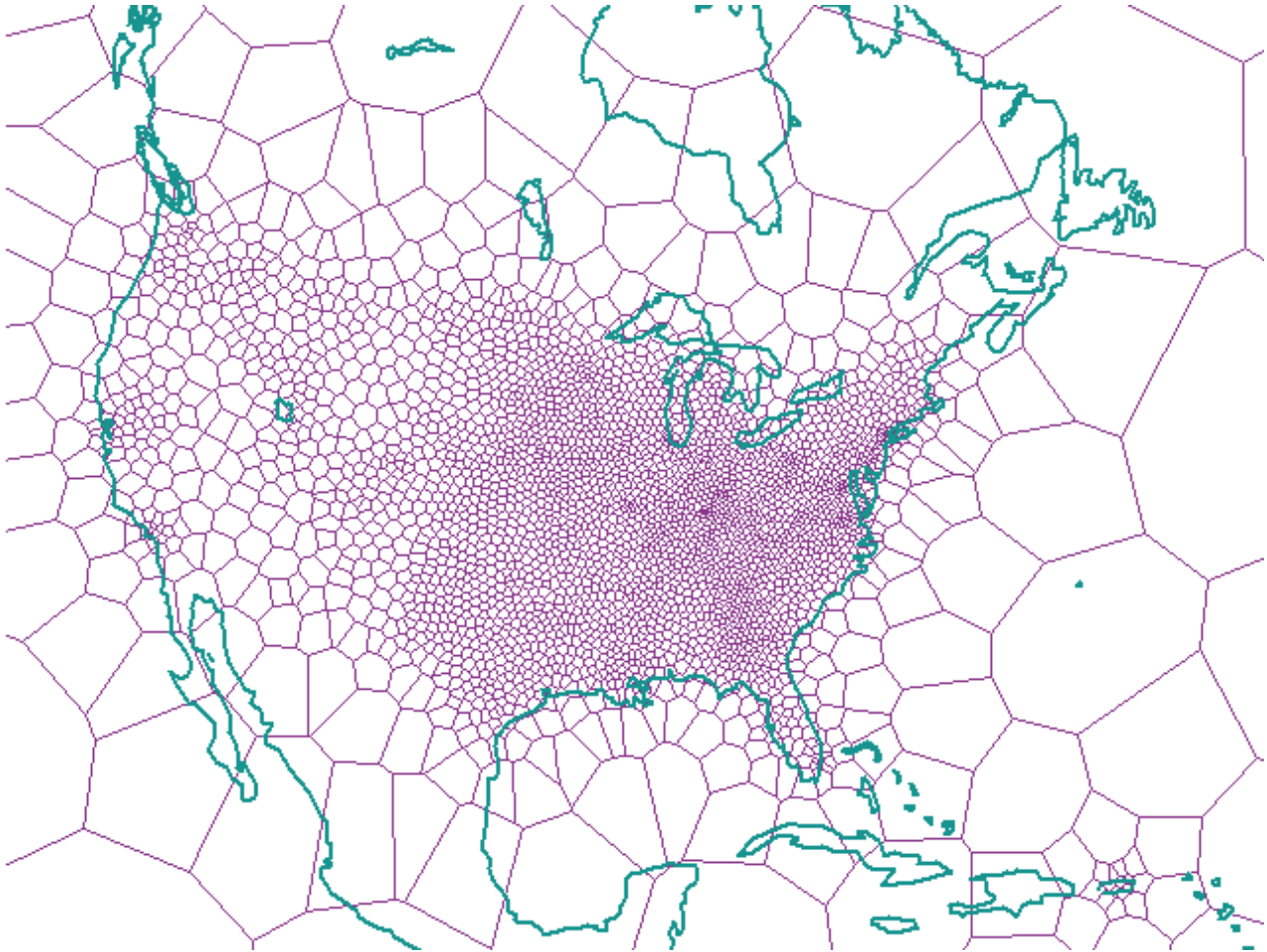


Figura 23. Struttura di celle Voronoi per gli USA

Canada (ID Voronoi: 3)

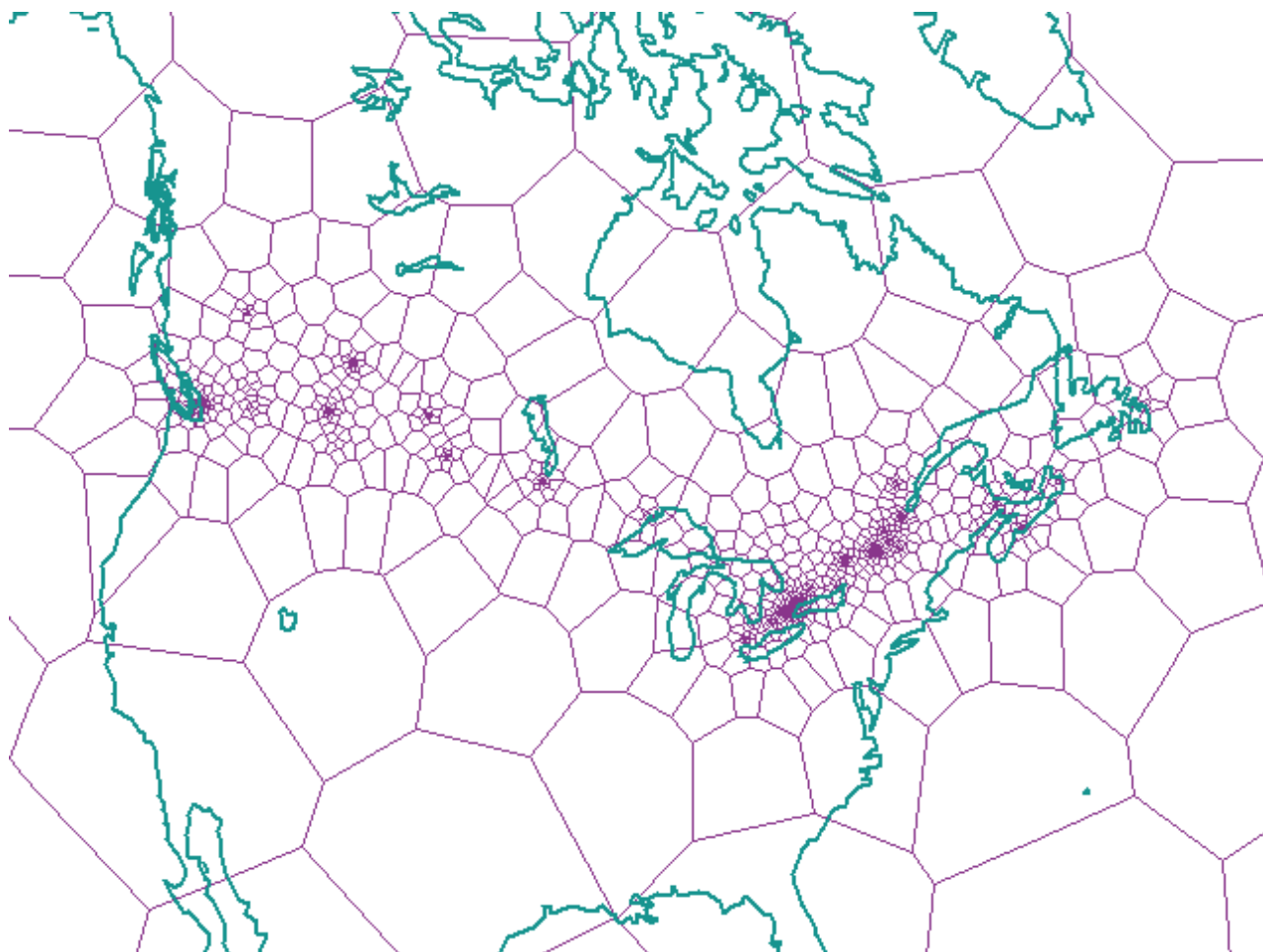


Figura 24. Struttura di celle Voronoi per il Canada

India (ID Voronoi: 4)

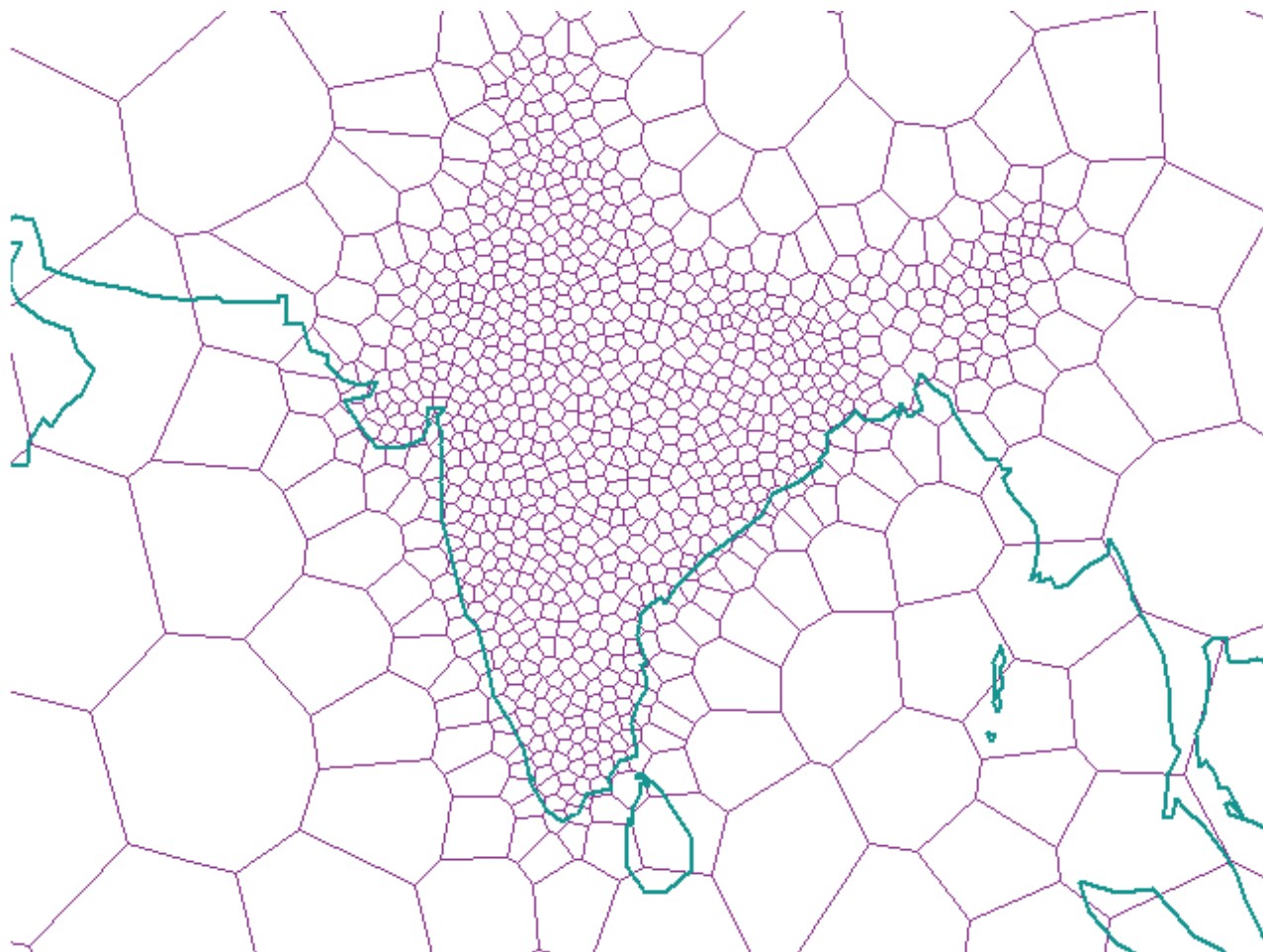


Figura 25. Struttura di celle Voronoi per l'India

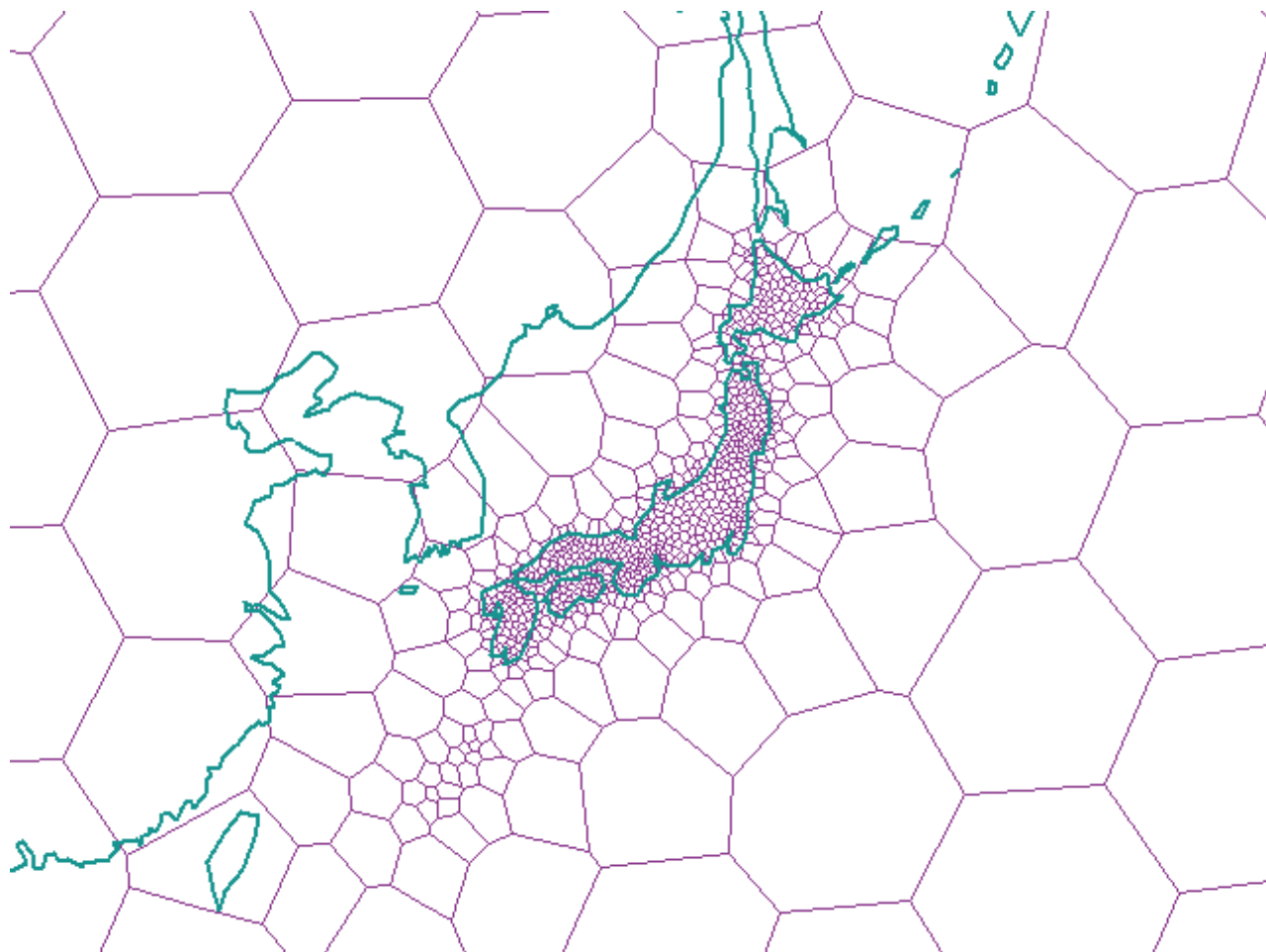
Giappone (ID Voronoi: 5)

Figura 26. Struttura di celle Voronoi per il Giappone

Africa (ID Voronoi: 6)

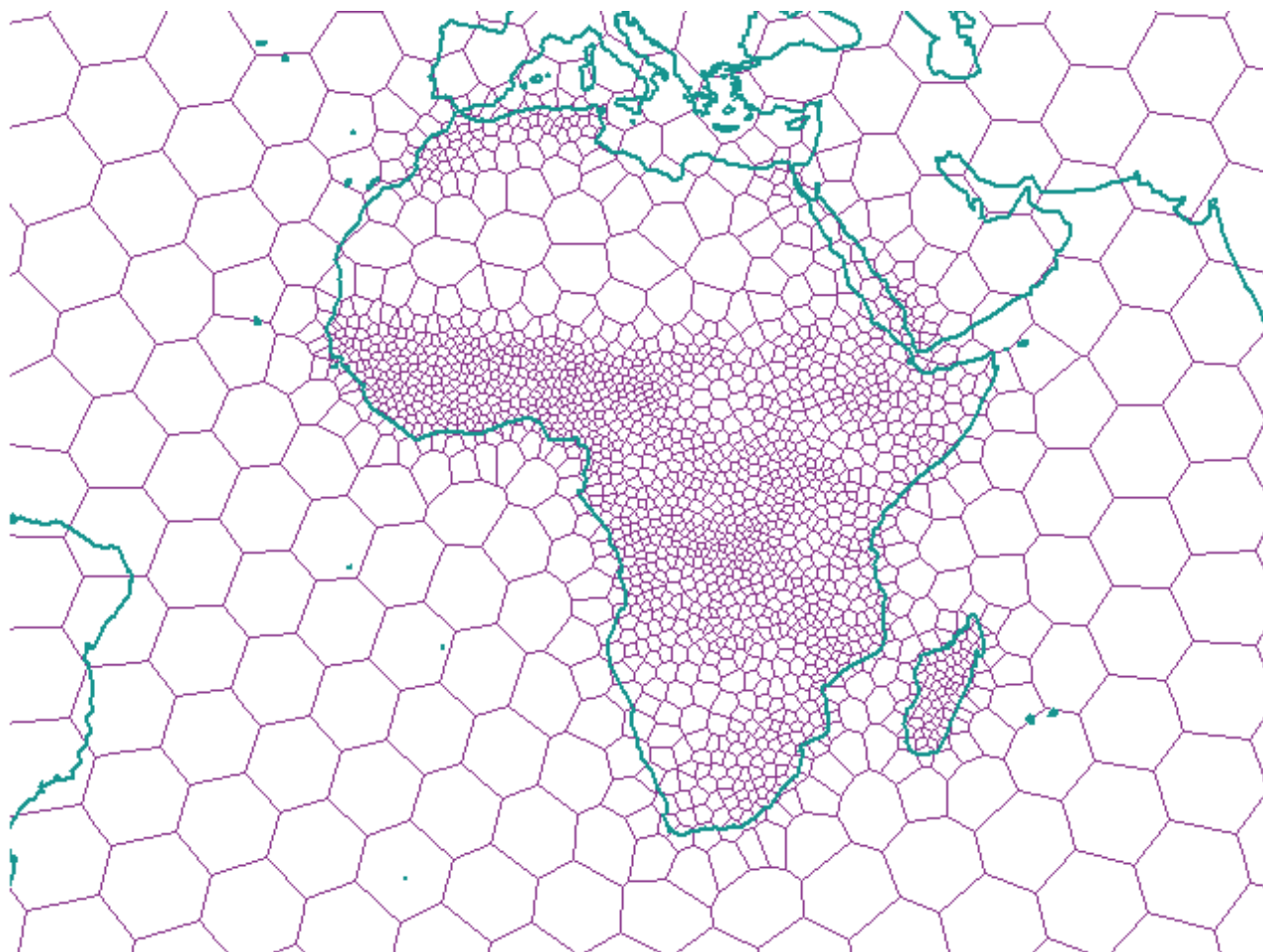


Figura 27. Struttura di celle Voronoi per l'Africa

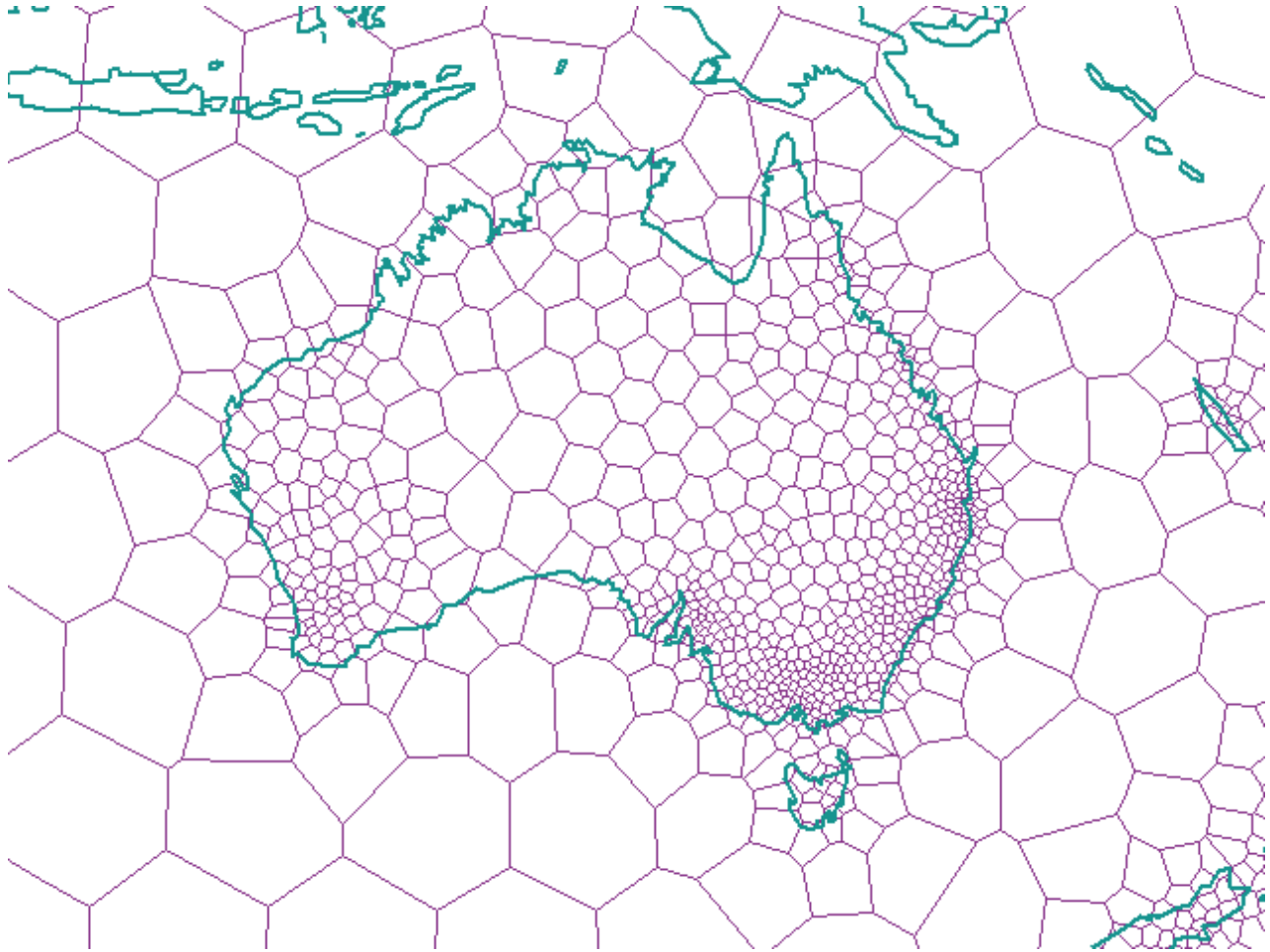
Australia (ID Voronoi: 7)

Figura 28. Struttura di celle Voronoi per l'Australia

Europa (ID Voronoi: 8)

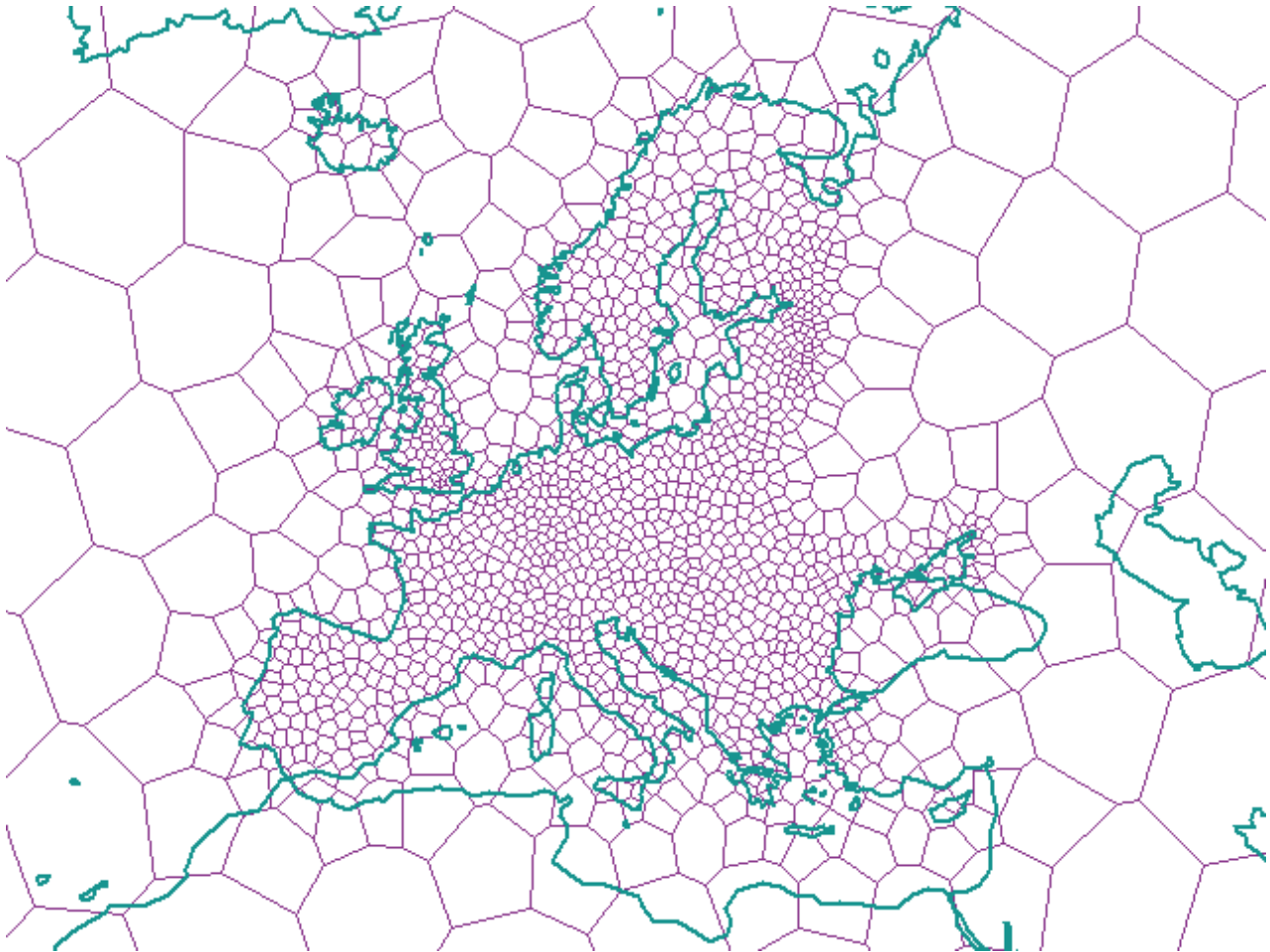


Figura 29. Struttura di celle Voronoi per l'Europa

America del Nord (ID Voronoi: 9)

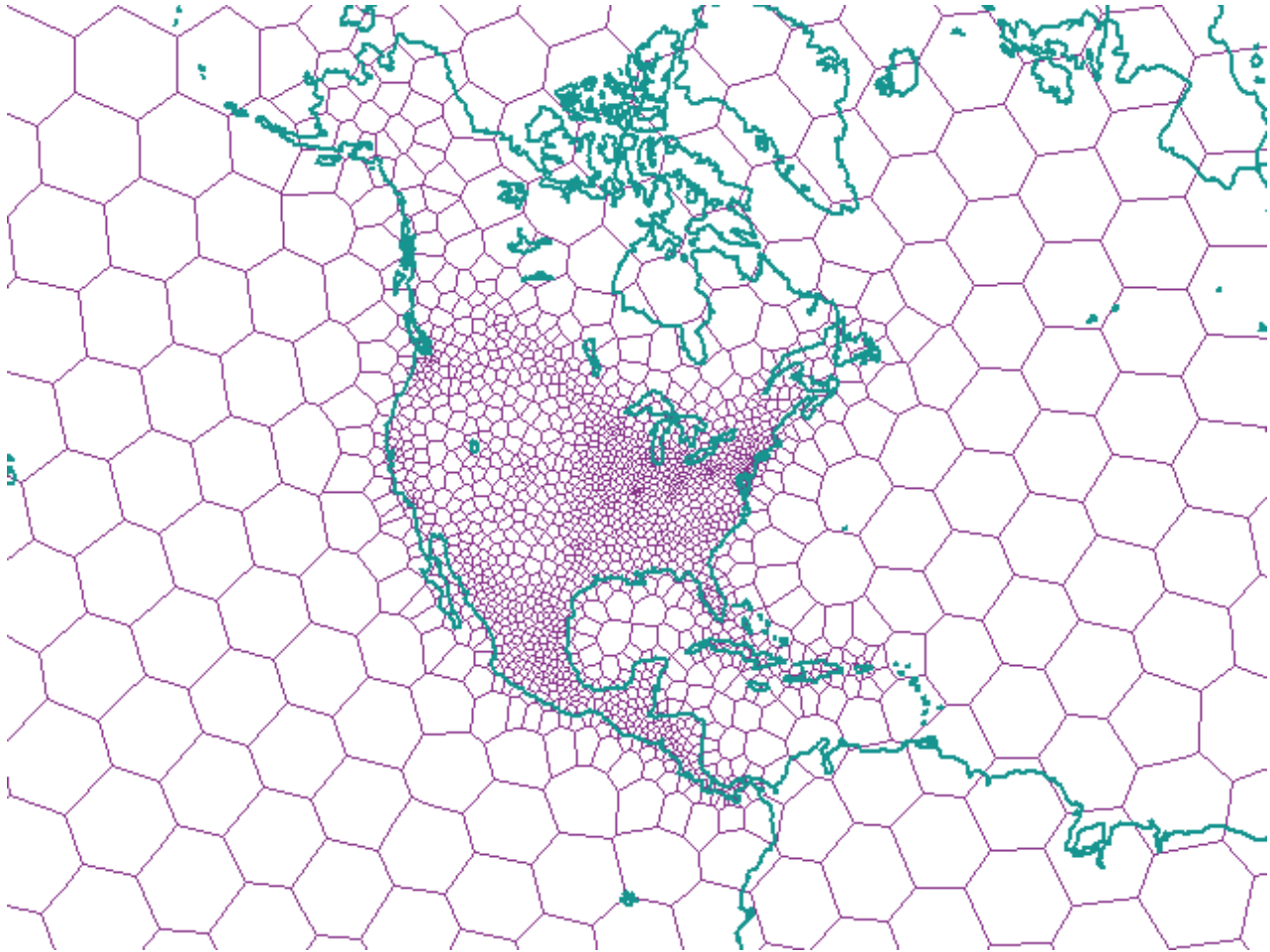


Figura 30. Struttura di celle Voronoi per l'America del Sud

America del Sud (ID Voronoi: 10)



Figura 31. Struttura di celle Voronoi per l'America del Sud

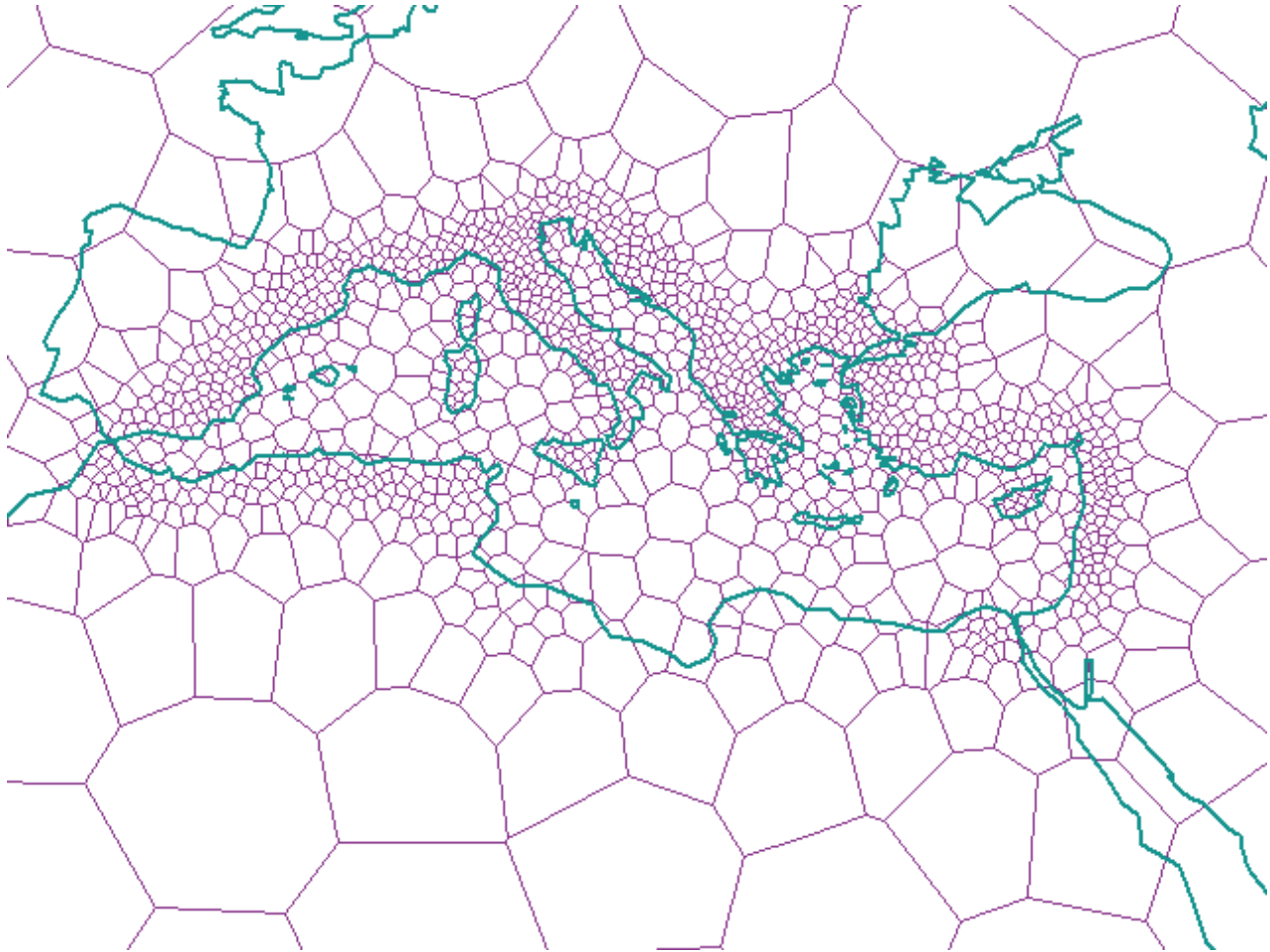
Mediterraneo (ID Voronoi: 11)

Figura 32. Struttura di celle Voronoi per il Mediterraneo

**Mondiale, distribuzione dati uniforme, risoluzione media –
dodeca04 (ID Voronoi: 12)**



Figura 33. Struttura di celle Voronoi mondiale (dodeca04)

Mondiale, nazioni industriali – nazioni G7 (Id Voronoi: 13)

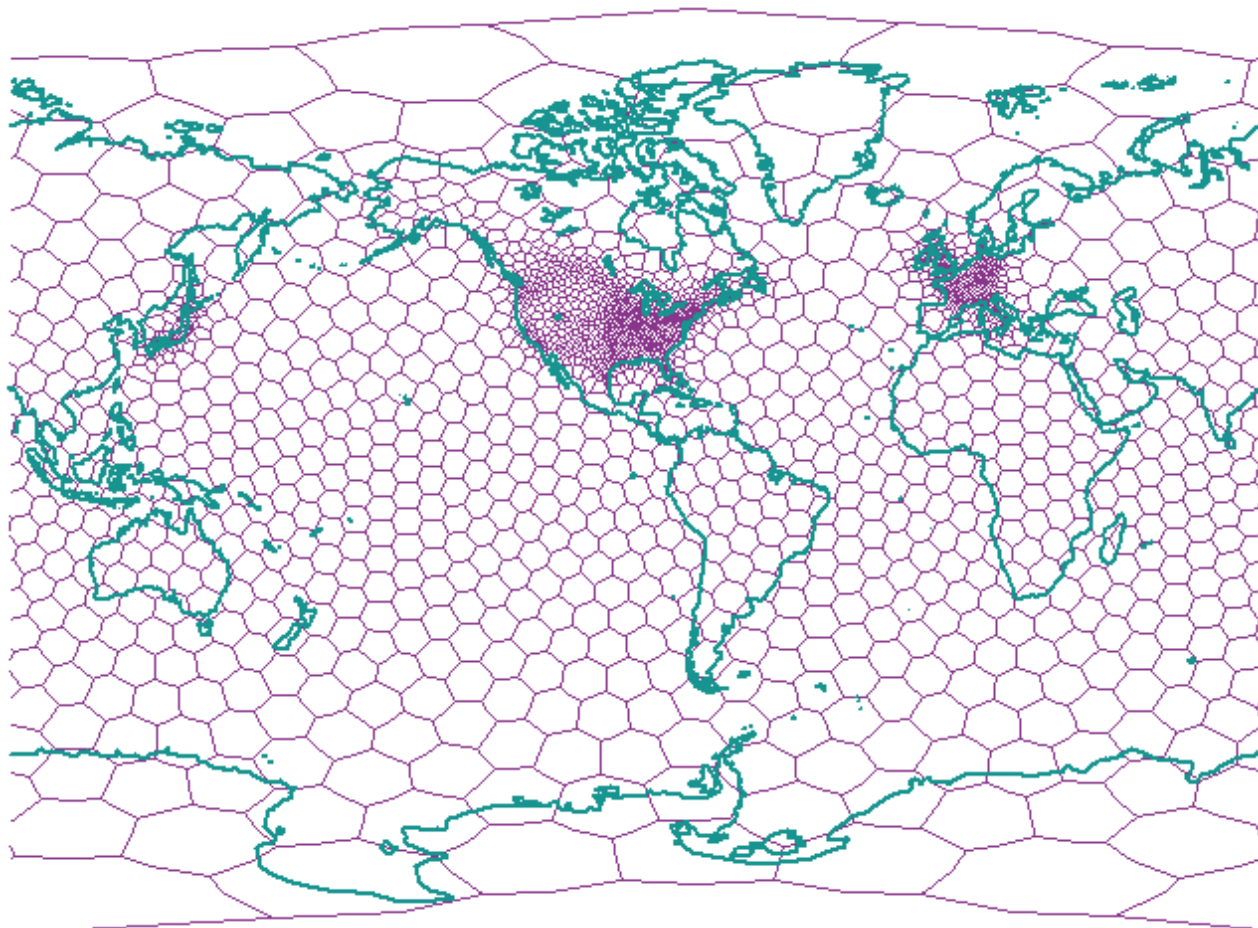


Figura 34. Struttura di celle Voronoi per (g7nations)

Mondiale, distribuzione dati uniforme, risoluzione bassa – isotipo (ID Voronoi: 14)



Figura 35. Struttura di celle Voronoi mondiale (isotipo)

Capitolo 19. Differenze nell'utilizzo dei dati geodetici e spaziali

Questo capitolo descrive le seguenti differenze nell'uso dei dati geodetici e spaziali:

- Attributi *x* e *y* minimi e massimi per i tipi di dati ST_Geometry
- Differenze nell'uso di rappresentazioni a Terra piana e a Terra sferica
- Funzioni spaziali supportate da DB2 Geodetic Extender e differenze nel comportamento di tali funzioni.
- Procedure memorizzate e viste di catalogo supportate da DB2 Geodetic Extender
- Altri sistemi di riferimento spaziali (datum) e ellissoidi geodetici

Attributi *x* e *y* minimi e massimi

Per organizzare i dati in strutture di celle per l'indice geodetico Voronoi, DB2[®] Geodetic Extender utilizza un MBC (minimum bounding circle) invece di un MBR (minimum bounding rectangle).

Per le forme geometriche geodetiche, l'MBC è un cerchio che circonda le forme e i valori interni minimi e massimi di *x* e *y* saranno i seguenti:

xmin Il termine *i* del coseno della direzione del centro del cerchio.

xmax Il termine *j* del coseno della direzione del centro del cerchio.

ymin Il termine *k* del coseno della direzione del centro del cerchio.

ymax L'*arc_radius* del cerchio.

Per le forme geometriche geodetiche, le funzioni ST_MinX, ST_MaxX, ST_MinY e ST_MaxY mostrano i punti dell'MBC. I risultati di tali funzioni producono tuttavia valori di latitudine e longitudine simili a quelli delle forme geometriche spaziali, ma i risultati per le forme geometriche geodetiche possono risultare diverse nel modo seguente::

- Se MBC attraversa la linea, il valore di ST_MinX sarà maggiore del valore di ST_MaxX. Ad esempio, se il centro di un MBC si trova sulla linea e ha un raggio di 5 gradi, il valore di ST_MinX sarà 175 e il valore di ST_MaxX sarà -175.
- Se l'MBC comprende il Polo Nord e Sud, ST_MinX sarà -180 e ST_MaxX 180.
- Se l'MBC include il Polo Nord, il valore di ST_MaxY è 90.
- Se l'MBC include il Polo Sud, il valore di ST_MinY sarà -90.

Differenze nell'uso di rappresentazioni della Terra piana o sferica

DB2 Spatial Extender e DB2 Geodetic Extender utilizzano tecnologie diverse:

- Spatial Extender utilizza una carta geografica piana, basata su coordinate proiettate. Tuttavia, nessuna proiezione può rappresentare fedelmente l'intero pianeta in quanto tutte le carte geografiche hanno un limite mentre la Terra non ne ha.
- Geodetic Extender utilizza un ellissoide come modello della Terra senza distorsioni ai poli o interruzioni al 180° meridiano.

Differenze nell'utilizzo dei dati geodetici e spaziali

In questa sezione, il termine "Terra-piana" si riferisce all'uso di una proiezione per rappresentare l'intero pianeta. Il termine "Terra-sferica" si riferisce all'uso di un sistema di riferimento che utilizza un ellissoide come modello della Terra.

Le diverse tecnologie portano a una diversa gestione delle forme geometriche in alcune situazioni, soprattutto quelle descritte in questa sezione:

- Segmenti di linee (e distanze misurate) che attraversano il 180° meridiano.
- Poligoni che si trovano a cavallo del 180° meridiano.
- MBR che attraversano il 180° meridiano.
- Poligoni che contengono un polo.
- Poligoni che rappresentano gli emisferi, le cinture equatoriali o l'intero pianeta.

Geodetic Extender ha un vantaggio particolare quando si lavora con forme geometriche che attraversano il 180° meridiano, o sono vicine a un polo, mentre la rappresentazione Terra-piana utilizzata da Spatial Extender ha delle limitazioni.

Segmenti di linee che attraversano il 180° meridiano

La Figura 36 a pagina 199 mostra i diversi modi in cui Spatial Extender e Geodetic Extender gestiscono un segmento di linea che attraversa il 180° meridiano. In questo esempio, il segmento di linea viene utilizzato per misurare la distanza tra Anchorage e Tokyo. Geodetic Extender misura la distanza tra due punti di un geodesico, la distanza più breve tra due punti dell'ellissoide (consultare la sezione "Distanze geodesiche" a pagina 160). I due punti possono trovarsi in un punto qualsiasi della Terra; Geodetic Extender sceglierà correttamente un segmento in direzione ovest da Anchorage a Tokyo perché utilizza la rappresentazione Terra-sferica. Invece, Spatial Extender, poiché utilizza una proiezione piana, non sarà consapevole che un segmento può unire Anchorage e Tokyo in questo modo e sceglierà un segmento molto più lungo in direzione est verso Tokyo. La proiezione piana, ha il meridiano -180° come margine sinistro e il meridiano 180° come margine destro.

Per ottenere un risultato corretto con Spatial Extender, è necessario procedere in uno dei seguenti modi:

- Dividere il segmento in due, uno a est del 180° meridiano e uno a ovest.
- Riproiettare i dati in modo che il 180° meridiano non risulti più un margine.

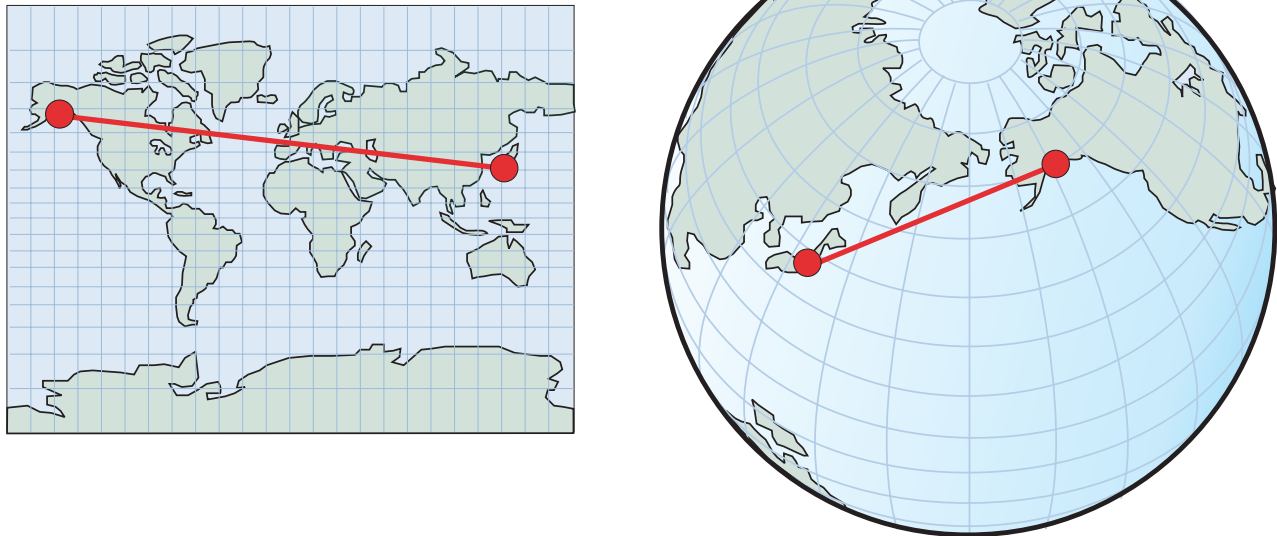


Figura 36. Linee che attraversano il 180° meridiano

Poligoni a cavallo del 180° meridiano

Per gestire un poligono a cavallo del 180° meridiano nella rappresentazione Terra-piana (Spatial Extender), sarà necessario dividere il poligono in due parti, una per la parte a est del 180° meridiano e l'altra a ovest:

```
MULTIPOLYGON(
  ((-180 30, -165 30, -165 40, -180 40, -180 30)),
  ((180 30, 180 40, 165 40, 165 30, 180 30)))
```

Come si evince dalla figura Figura 37 a pagina 200, nella rappresentazione Terra-sferica (Geodetic Extender) non è necessaria tale divisione, in quanto è possibile utilizzare un singolo poligono inalterato:

```
POLYGON((165 30, -165 30, -165 40, 165 40, 165 30))
```

Differenze nell'utilizzo dei dati geodetici e spaziali

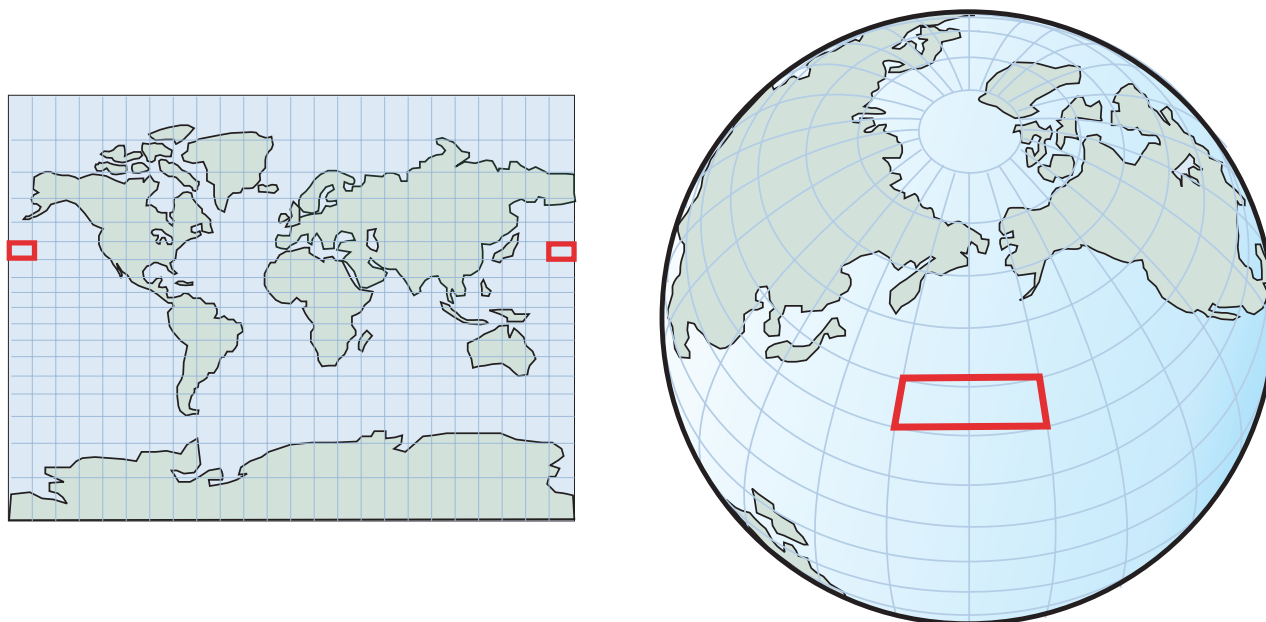


Figura 37. Poligoni a cavallo del 180° meridiano—creazione di due poligoni separati

Se, utilizzando Spatial Extender, non vengono creati i due poligoni separati, il programma riordinerà i vertici del poligono in modo che definisca un'area diversa, come mostrato nella Figura 38 a pagina 201. La parte superiore della Figura 38 a pagina 201 mostra i vertici corretti di un poligono a cavallo del 180° meridiano:

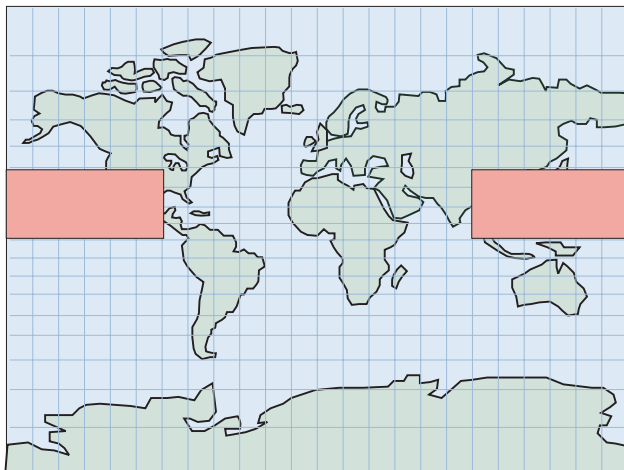
```
POLYGON((90 0, -90 0, -90 40, 90 40, 90 0))
```

La parte inferiore della Figura 38 a pagina 201 mostra i vertici riordinati che risultano in un poligono che non giace più sul 180° meridiano, ma sul meridiano 0.

```
POLYGON((-90 0, 90 0, 90 40, -90 40, -90 0))
```

Si desidera un poligono a cavallo del 180° meridiano:

Poligono ((90 0, -90 0, -90 40, 90 40))



Spatial Extender riordina i vertici ed il poligono risultante definisce un'area diversa:

Poligono ((-90 0, 90 0, 90 40, -90 40))

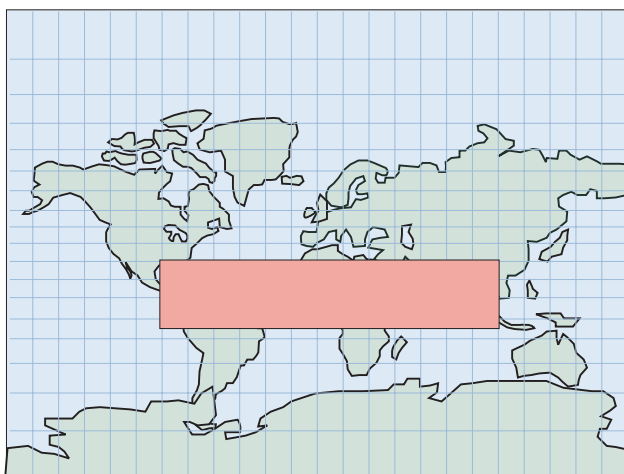


Figura 38. Vertici riordinati dei poligoni a cavallo del 180° meridiano

L'area definita sarà complementare all'area della Terra, e non all'area desiderata, come mostrato nella Figura 39 a pagina 202. In modo simile al segmento nell'esempio precedente, un altro modo di gestire questa situazione consiste nel riproiettare i dati in modo che il 180° meridiano non rappresenti un margine.

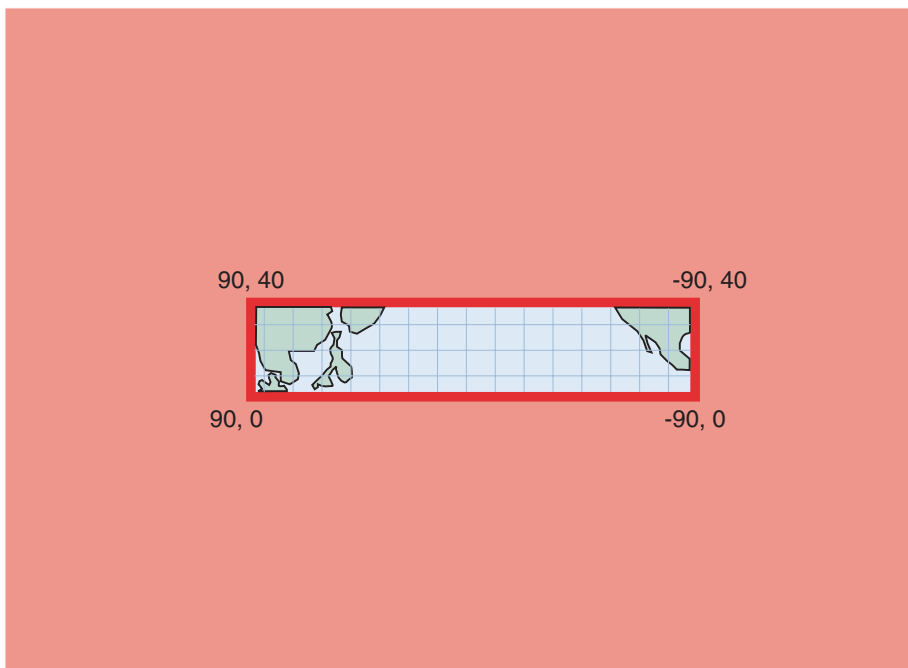


Figura 39. Poligoni che si trovano a cavallo del 180° meridiano

Poligoni che contengono un polo

La Figura 40 a pagina 203 mostra come utilizzare un poligono contenente il Polo Sud con Spatial Extender o Geodetic Extender. Poiché utilizzando Spatial Extender, i dati utilizzati si trovano proprio al limite della proiezione piana, a causa della distorsione della carta geografica della superficie della Terra, sarà necessario aggiungere nuovi limiti e vertici per rappresentare il polo in un poligono:

```
POLYGON((-180 -90, 180 -90, 180 -60, -180 -60, -180 -90))
```

La rappresentazione Terra-sferica (Geodetic Extender) mostra il poligono che circonda il Polo Sud come un cerchio che segue il parallelo -60° :

```
POLYGON((0 -60, -1 -60, -2 -60, ..., -179 -60, 180 -60, 179 -60, ..., 1 -60, 0 -60))
```

Un modo migliore per rappresentare questo cerchio, consiste nel riproiettare i dati in modo che l'intero Polo Sud e l'area circostante siano visibili sulla carta geografica.

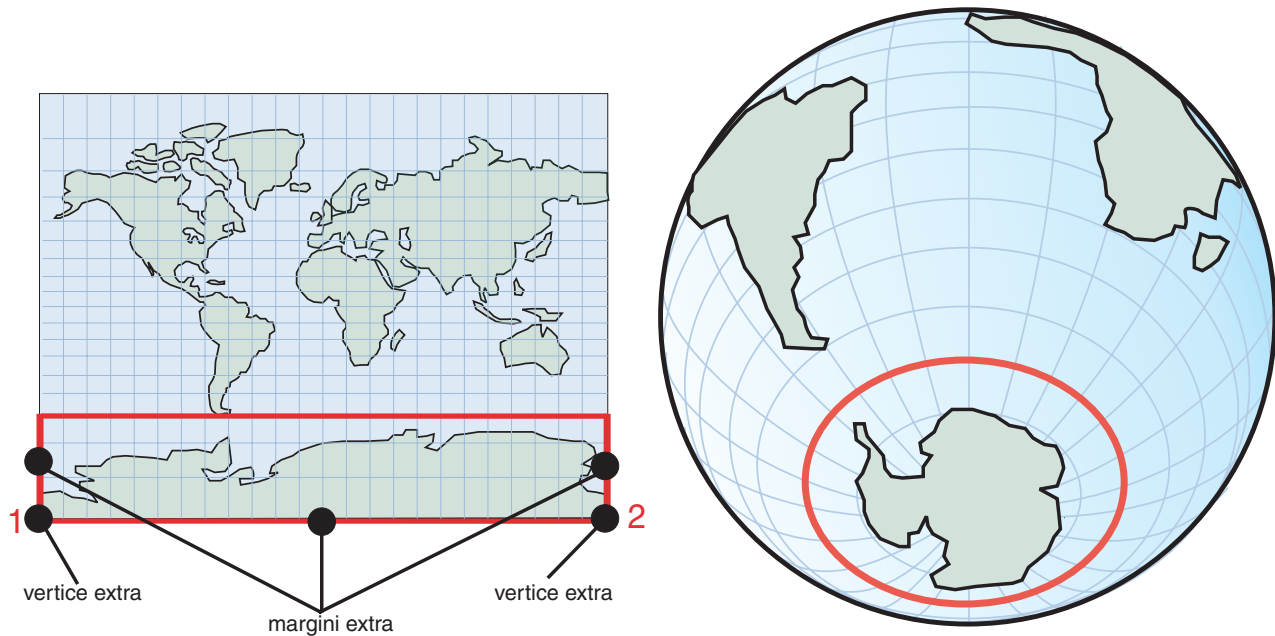


Figura 40. Poligoni che contengono un polo

Negli esempi precedenti, scegliendo un sistema di riferimento spaziale appropriato, è possibile ottenere risultati precisi. Tuttavia, nessuna proiezione è in grado di risolvere tutti gli esempi contemporaneamente. Ad esempio, una proiezione che non ha il 180° meridiano come margine, sposterà il margine in un'altra posizione ed eviterà l'area che presenta problemi.

Poligoni che rappresentano gli emisferi, le cinture equatoriali e l'intero pianeta

Quando si desidera utilizzare un poligono che rappresenta vaste aree della superficie terrestre, ad esempio uno degli emisferi, le cinture equatoriali o l'intero pianeta, è necessario conoscere in che modo Spatial Extender e Geodetic Extender gestiscono queste situazioni. In questi casi, la rappresentazione Terra-sferica ottiene risultati accurati per calcoli di aree e distanze, laddove non è possibile con una proiezione.

Ad esempio, la Figura 41 a pagina 204 mostra i poligoni che definiscono l'emisfero ovest in una rappresentazione Terra-piana (Spatial Extender) e in una rappresentazione Terra-sferica (Geodetic Extender).

- Nella rappresentazione Terra-piana, nella parte superiore della Figura 41 a pagina 204, quattro coordinate rappresentano l'emisfero occidentale nel formato WKT, ovvero 'POLYGON((0 -90, 0 90, -180 90, 180 -90, 0 -90))'.
- Nella rappresentazione Terra-sferica, quattro coordinate rappresentano l'emisfero occidentale nel formato WKT, ovvero 'POLYGON((0 0, 0 90, 180 0, 0 -90, 0 0))'. Queste quattro coordinate definiscono una circonferenza attorno alla Terra lungo il meridiano 0 e la rispettiva linea antipodale, il 180° meridiano.

Quando vengono specificati gli stessi quattro punti nell'ordine opposto, viene definito l'emisfero orientale.

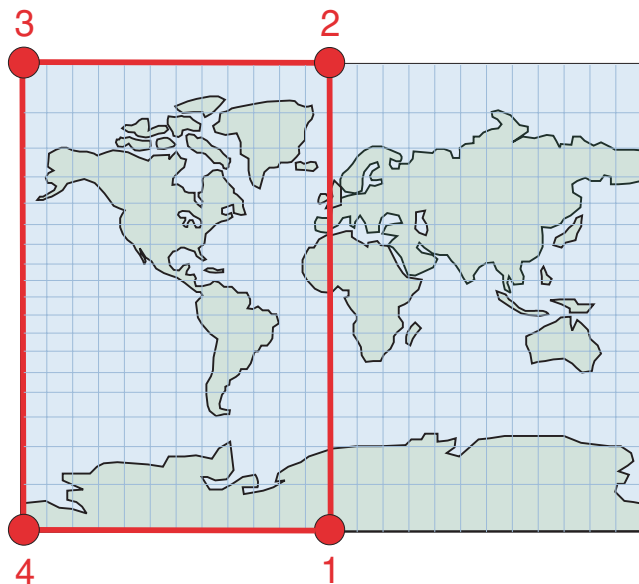
- In una rappresentazione piana, l'emisfero orientale è 'POLYGON((0 -90, 180 -90, 180 90, 0 90, 0 -90))'.

Differenze nell'utilizzo dei dati geodetici e spaziali

- In una rappresentazione sferica, l'emisfero orientale è 'POLYGON((0 -90, 180 0, 0 90, 0 0, 0 -90))'.

Emisfero occidentale, rappresentazione piana

Poligono ((0 -90, 0 90, -180 90, 180 -90, 0 -90))



Emisfero occidentale, rappresentazione sferica

Poligono ((0 0, 0 90, 180 0, 0 -90, 0 0))

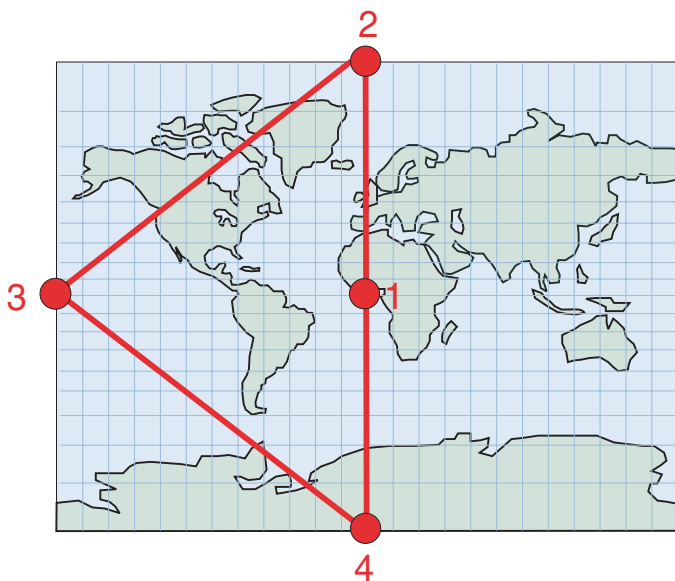


Figura 41. Poligoni che rappresentano gli emisferi

La Figura 42 a pagina 205, mostra le coordinate dei poligoni che definiscono la cintura equatoriale in una rappresentazione piana (Spatial Extender) e sferica (Geodetic Extender).

- La parte superiore della Figura 42 a pagina 205 mostra la rappresentazione piana della cintura equatoriale con coordinate in formato WKT, ovvero 'POLYGON((180 -60, 180 60, -180 60, -180 -60, 180 -60))'.

Differenze nell'utilizzo dei dati geodetici e spaziali

- Nella rappresentazione sferica nella parte inferiore della Figura 42, per rappresentare la cintura equatoriale, viene definita l'area di esclusione di due circonferenze:

```
'MULTIPOLYGON(((0 60, -120 60, 120 60, 0 60)),  
((0 -60, 120 -60, -120 -60, 0 -60)))'
```

Per maggiore chiarezza vengono mostrati solo tre punti in ciascuna circonferenza. In realtà, se si desidera avvicinare di più le cinture alla latitudine 60 o -60, sarà necessario aggiungere più punti intermedi. La prima circonferenza ((0 60, -120 60, 120 60, 0 60)) indica i vertici nell'ordine che definisce l'area sud della 60° linea di latitudine. La seconda circonferenza ((0 -60, 120 -60, -120 -60, 0 -60)) indica l'area nord della -60° linea di latitudine.

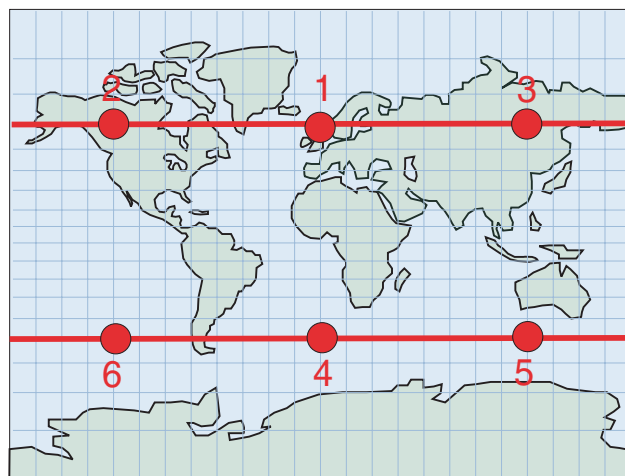
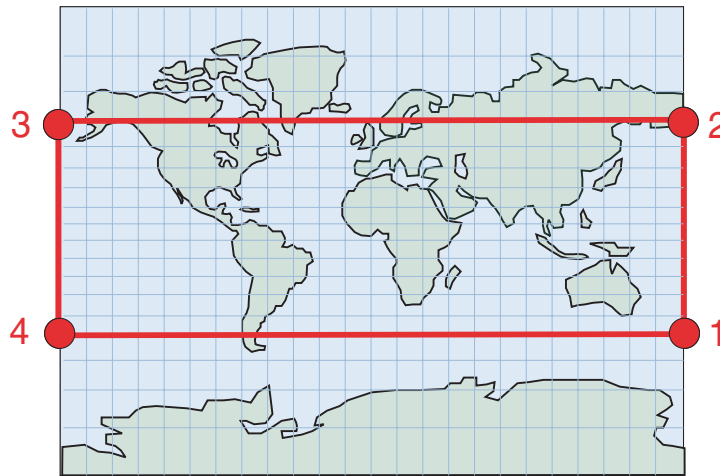
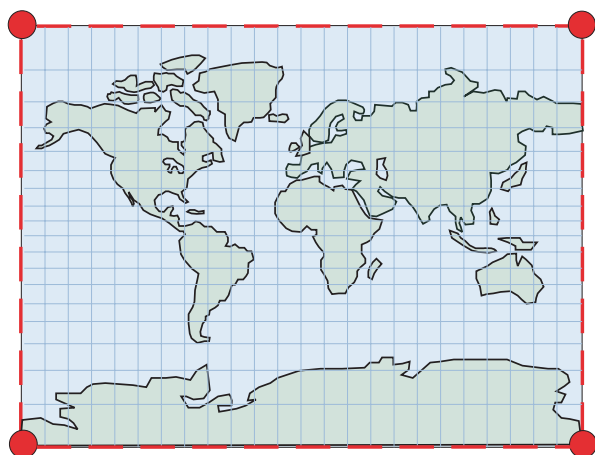


Figura 42. Poligoni che rappresentano le cinture equatoriali

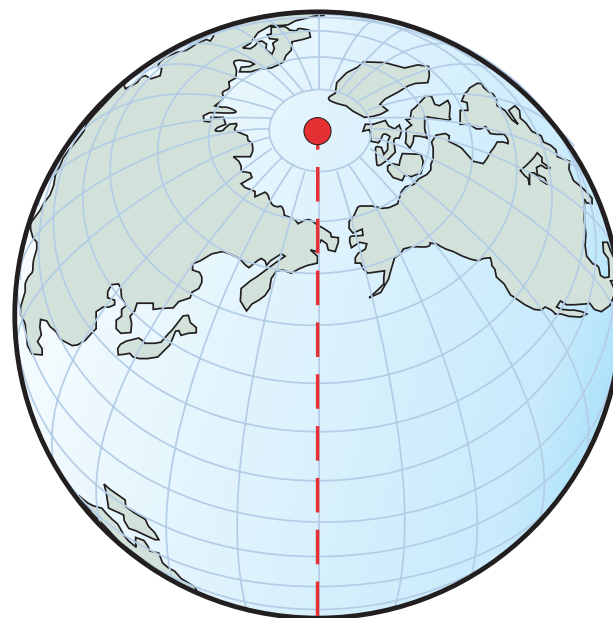
La Figura 43 a pagina 206, mostra i poligoni che definiscono l'intero pianeta in una rappresentazione piana (Spatial Extender) e sferica (Geodetic Extender). In

Differenze nell'utilizzo dei dati geodetici e spaziali

entrambi i casi la Terra viene rappresentata con lo stesso poligono nel formato WKT, ovvero 'POLYGON((-180 -90, 180 -90, 180 90, -180 90, -180 -90))'.



Rappresentazione piana



Rappresentazione ellissoidale.
Questo poligono non ha limite, quindi
sono necessarie informazioni speciali.

Figura 43. Poligoni che rappresentano l'intero pianeta

Argomenti correlati:

- "Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender" a pagina 158
- "Regioni geodetiche" a pagina 161
- "Latitudine e longitudine geodetiche" a pagina 159
- "Distanze geodesiche" a pagina 160
- "Sferoidi geodetici" a pagina 220

Funzioni spaziali supportate da DB2 Geodetic Extender

DB2 Spatial Extender è stato creato sulla libreria di funzioni fornita da ESRI, mentre DB2 Geodetic Extender sulla libreria di funzioni Hipparchus. Le differenze nella funzionalità delle librerie ESRI e Hipparchus non sono molte rispetto al comportamento di alcune funzioni. La seguente tabella mostra le funzioni di Spatial Extender supportate da Geodetic Extender e riporta le differenze di comportamento. Per informazioni sull'uso e sulla sintassi delle funzioni spaziali, consultare l'argomento relativo alla funzione spaziale appropriata.

Tabella 26. Supporto funzioni per Geodetic Extender

Funzione	E' supportata da DB2 Geodetic Extender?	Differenze nel comportamento per DB2 Geodetic Extender
EnvelopesIntersect	Sì	Nessuno
Aggregazione MBR	No	Non applicabile

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 26. Supporto funzioni per Geodetic Extender (Continua)

Funzione	E' supportata da DB2 Geodetic Extender?	Differenze nel comportamento per DB2 Geodetic Extender
ST_AppendPoint	No	Non applicabile
ST_Area	Sì	L'unità di misura predefinita è il metro.
ST_AsBinary	Sì	Nessuno
ST_AsGML	Sì	Nessuno
ST_AsShape	Sì	Nessuno
ST_AsText	Sì	Nessuno
ST_Boundary	No	Non applicabile
ST_Buffer	Sì	Supportata solo con punti e multipunti. La distanza può essere un valore negativo. L'unità di misura predefinita è il metro.
ST_Centroid	No	Non applicabile
ST_ChangePoint	No	Non applicabile
ST_Contains	Sì	Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.
ST_ConvexHull	No	Non applicabile
ST_CoordDim	Sì	Nessuno
ST_Crosses	No	Non applicabile
ST_Difference	Sì	Non supportato con linee e multilinee. Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico. La dimensione della forma geometrica restituita è uguale alla dimensione delle forme geometriche di input.
ST_Dimension	Sì	Nessuno
ST_Disjoint	Sì	Nessuno
ST_Distance	Sì	Restituisce la <i>distanza geodesica</i> . Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico. L'unità di misura predefinita è il metro.
ST_Edge_GC_USA	Sì	Nessuno
ST_Endpoint	Sì	Nessuno
ST_Envelope	Sì	L'involuppo è un poligono che racchiude l'MBC della forma geometrica.
ST_EnvIntersects	Sì	Nessuno
ST_EqualCoordsys	Sì	Nessuno
ST_Equals	No	Non applicabile
ST_EqualSRS	Sì	Nessuno
ST_ExteriorRing	Sì	Nessuno
ST_FindMeasure o ST_LocateAlong	No	Non applicabile
ST_Generalize	Sì	L'unità per la soglia è il metro.
ST_GeomCollection	No	Non applicabile
ST_GeomCollFromTxt	No	Non applicabile

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 26. Supporto funzioni per Geodetic Extender (Continua)

Funzione	E' supportata da DB2 Geodetic Extender?	Differenze nel comportamento per DB2 Geodetic Extender
ST_GeomCollFromWKB	No	Non applicabile
ST_Geometry	Sì	Nessuno
ST_GeometryN	Sì	Nessuno
ST_GeometryType	Sì	Nessuno
ST_GeomFromText	Sì	Nessuno
ST_GeomFromWKB	Sì	Nessuno
ST_GetIndexParms	No	Non applicabile
ST_InteriorRingN	Sì	Nessuno
ST_Intersection	Sì	La dimensione della forma geometrica restituita corrisponde a quella dell'input di dimensioni inferiori, ad eccezione della dimensione dell'intersezione di due linee.
ST_Intersects	Sì	Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
ST_Is3d	Sì	Nessuno
ST_IsClosed	Sì	Nessuno
ST_IsEmpty	Sì	Nessuno
ST_IsMeasured	Sì	Nessuno
ST_IsRing	No	Non applicabile
ST_IsSimple	No	Non applicabile
ST_IsValid	Sì	Nessuno
ST_Length	Sì	L'unità di misura predefinita è il metro.
ST_LineFromText	Sì	Nessuno
ST_LineFromWKB	Sì	Nessuno
ST_LineString	Sì	Nessuno
ST_LineStringN	Sì	Nessuno
ST_M	Sì	Nessuno
ST_MaxM	Sì	Nessuno
ST_MaxX	Sì	Restituisce il valore X massimo dell'MBC (minimum bounding circle). Nota: Se MBC attraversa la linea, il valore di ST_MaxX sarà inferiore al valore di ST_MinX. Se l'MBC comprende il Polo Nord e Sud, ST_MinX sarà -180 e ST_MaxX 180.
ST_MaxY	Sì	Restituisce il valore massimo Y dell'MBC. Nota: Se l'MBC include il Polo Nord, il valore di ST_MaxY è 90.
ST_MaxZ	Sì	Nessuno
ST_MBR	Sì	L'MBR è una forma geometrica che racchiude l'MBC della forma geometrica.
ST_MBRIntersects	Sì	Nessuno

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 26. Supporto funzioni per Geodetic Extender (Continua)

Funzione	E' supportata da DB2 Geodetic Extender?	Differenze nel comportamento per DB2 Geodetic Extender
ST_MeasureBetween o ST_LocateBetween	No	Non applicabile
ST_MidPoint	Sì	Nessuno
ST_MinM	Sì	Nessuno
ST_MinX	Sì	Restituisce il valore minimo X dell'MBC. Nota: Se MBC attraversa la linea, il valore di ST_MinX sarà maggiore del valore di ST_MaxX. Se l'MBC comprende il Polo Nord e Sud, ST_MinX sarà -180 e ST_MaxX 180.
ST_MinY	Sì	Restituisce il valore minimo Y dell'MBC. Nota: Se l'MBC include il Polo Sud, il valore di ST_MinY sarà -90.
ST_MinZ	Sì	Nessuno
ST_MLineFromText	Sì	Nessuno
ST_MLineFromWKB	Sì	Nessuno
ST_MPointFromText	Sì	Nessuno
ST_MPointFromWKB	Sì	Nessuno
ST_MPolyFromText	Sì	Nessuno
ST_MPolyFromWKB	Sì	Nessuno
ST_MultiLineString	Sì	Nessuno
ST_MultiPoint	Sì	Nessuno
ST_MultiPolygon	Sì	Nessuno
ST_NumGeometries	Sì	Nessuno
ST_NumInteriorRing	Sì	Nessuno
ST_NumLineStrings	Sì	Nessuno
ST_NumPoints	Sì	Nessuno
ST_NumPolygons	Sì	Nessuno
ST_Overlaps	No	Non applicabile
ST_Perimeter	Sì	L'unità di misura predefinita è il metro.
ST_PerpPoints	No	Non applicabile
ST_Point	Sì	Nessuno
ST_PointFromText	Sì	Nessuno
ST_PointFromWKB	Sì	Nessuno
ST_PointN	Sì	Nessuno
ST_PolyFromText	Sì	Nessuno
ST_PolyFromWKB	Sì	Nessuno
ST_PointOnSurface	Sì	Nessuno
ST_Polygon	Sì	Nessuno
ST_PolygonN	Sì	Nessuno
ST_Relate	No	Non applicabile
ST_RemovePoint	No	Non applicabile

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 26. Supporto funzioni per Geodetic Extender (Continua)

Funzione	E' supportata da DB2 Geodetic Extender?	Differenze nel comportamento per DB2 Geodetic Extender
ST_SrsId or ST_SRID	Sì	Nessuno
ST_SrsName	Sì	Nessuno
ST_StartPoint	Sì	Nessuno
ST_SymDifference	Sì	Non supportato con linee e multilinee. La dimensione della forma geometrica restituita è uguale alla dimensione delle forme geometriche di input. Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
ST_ToGeomColl	No	Non applicabile
ST_ToLineString	Sì	Nessuno
ST_ToMultiLine	Sì	Nessuno
ST_ToMultiPoint	Sì	Nessuno
ST_ToPoint	Sì	Nessuno
ST_ToPolygon	Sì	Nessuno
ST_Touches	No	Non applicabile
ST_Transform	Sì	Nessuno. Nota: le trasformazioni delle coordinate vengono eseguite punto per punto. Durante la trasformazione tra i sistemi di coordinate geodetiche e sistemi di coordinate planari non proiettate, controllare attentamente eventuali poligoni e linee che si trovano a cavallo del 180° meridiano o che includono entrambi i poli. Poiché Spatial Extender e Geodetic Extender gestiscono questi casi in modo diverso, è possibile che le forme geometriche valide in un sistema di coordinate piane, non siano valide in un sistema di coordinate ovali e viceversa. Per ulteriori informazioni, consultare la sezione "Differenze nell'uso di rappresentazioni della Terra piana o sferica" a pagina 197.
ST_Union	Sì	Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
ST_Within	Sì	Le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
ST_WKBToSQL	Sì	Nessuno
ST_WKTTToSQL	Sì	Nessuno
ST_X	Sì	Nessuno
ST_Y	Sì	Nessuno
ST_Z	Sì	Nessuno
Aggregazione unione	No	Non applicabile

Argomenti correlati:

- “Quando utilizzare DB2 Geodetic Extender e DB2 Spatial Extender” a pagina 158
- “Regioni geodetiche” a pagina 161
- “Latitudine e longitudine geodetiche” a pagina 159
- “Distanze geodesiche” a pagina 160

Attività correlate:

- “Creazione di indici geodetici Voronoi” a pagina 179

Riferimenti correlati:

- “Differenze nell'uso di rappresentazioni della Terra piana o sferica” a pagina 197

Procedure memorizzate e viste del catalogo di DB2 Geodetic Extender

DB2 Geodetic Extender supporta le stesse viste del catalogo di DB2 Spatial Extender e un sottoinsieme delle procedure memorizzate spaziali.

Geodetic Extender non supporta le seguenti procedure memorizzate:

- ST_disable_autogeocoding
- ST_enable_autogeocoding
- ST_register_geocoder
- ST_remove_geocoding_setup
- ST_run_geocoding
- ST_setup_geocoding
- ST_unregister_geocoder

Geodetic Extender fornisce 318 sistemi di riferimento spaziali geodetici predefiniti, contenuti nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS. Consultare la sezione “Datum supportati da DB2 Geodetic Extender” per un elenco completo.

Datum supportati da DB2 Geodetic Extender

Come descritto nella sezione “Sistema di coordinate geografiche” a pagina 59, un *datum* è una serie di valori che definisce la posizione di un ellissoide relativo al centro della terra. Un sistema di riferimento spaziale (SRS) è un insieme di parametri che associano un datum a un ellissoide e viene identificato mediante un identificativo SRID (spatial reference system identifier). La Tabella 28 elenca i datum forniti da DB2 Geodetic Extender. I valori di scarto e i fattori di scala per tutti i sistemi di riferimento spaziali geodetici predefiniti sono gli stesso, e nella tabella seguente sono mostrati i rispettivi valori.

Tabella 27. Valori di scarto e di scala per i sistemi di riferimento spaziali geodetici predefiniti

Parametro SRS	Valore
<i>xOffset</i>	-180
<i>yOffset</i>	-90
<i>zOffset</i>	-50000
<i>mOffset</i>	-1000
<i>xScale</i>	5965232
<i>yScale</i>	5965232
<i>zScale</i>	1000

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 27. Valori di scarto e di scala per i sistemi di riferimento spaziali geodetici predefiniti (Continua)

Parametro SRS	Valore
<i>mScale</i>	1000

yScale è sempre uguale a *xScale*.

E' possibile scegliere qualsiasi datum elencato nella Tabella 28 per il proprio sistema di riferimento spaziale. Scegliere quello più adatto ai dati utilizzati. Ad esempio, uno dei datum più comunemente utilizzato, World Geodetic System 1984 (WGS 1984), utilizza il centro della Terra come punto di origine e ricopre l'intero pianeta; viene definito datum centrato. Al contrario, un datum regionale, ad esempio il datum North American 1927, ricopre l'America del Nord a partire da un punto specifico. I dati regionali sono molto precisi per le regioni definite, ma il datum geodetico centrato è necessario per gestire le posizioni in tutto il pianeta.

Tabella 28. SRID con datum e ellissoide associati

SRID	Nome datum	Ellissoide di riferimento
2000000000	WGS 1984	WGS 1984
2000000001	Abidjan 1987	Clarke 1880 (RGS)
2000000002	Accra	War Office
2000000003	Adindan	Clarke 1880 (RGS)
2000000004	Afgooye	Krasovsky 1940
2000000005	Agadez	Clarke 1880 (IGN)
2000000006	Australian Geodetic Datum 1966	Australian
2000000007	Australian Geodetic Datum 1984	Australian
2000000008	Ain el Abd 1970	International 1924
2000000009	Airy 1830	Airy 1830
2000000010	Airy Modified	Airy Modified
2000000011	Alaskan Islands	Clarke 1866
2000000012	Amersfoort	Bessel 1841
2000000013	Anguilla 1957	Clarke 1880 (RGS)
2000000014	Anna 1 Astro 1965	Australian
2000000015	Antigua Astro 1943	Clarke 1880 (RGS)
2000000016	Aratu	International 1924
2000000017	Arc 1950	Clarke 1880 (Arc)
2000000018	Arc 1960	Clarke 1880 (RGS)
2000000019	Ascension Island 1958	International 1924
2000000020	Assumed Geographic (NAD27 per file shape senza PRJ)	Clarke 1866
2000000021	Astronomical Station 1952	International 1924
2000000022	ATF (Paris)	Plessis 1817
2000000023	Average Terrestrial System 1977	ATS 1977
2000000024	Australian National	Australian
2000000025	Ayabelle Lighthouse	Clarke 1880 (RGS)

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000026	Bab South Astro (Bablethuap Is, Republic of Palau)	Clarke 1866
2000000027	Barbados 1938	Clarke 1880 (RGS)
2000000028	Batavia	Bessel 1841
2000000029	Batavia (Jakarta)	Bessel 1841
2000000030	Astro Beacon E 1945	International 1924
2000000031	Beduaram	Clarke 1880 (IGN)
2000000032	Beijing 1954	Krasovsky 1940
2000000033	Reseau National Belge 1950	International 1924
2000000034	Belge 1950 (Brussels)	International 1924
2000000035	Reseau National Belge 1972	International 1924
2000000036	Bellevue (IGN)	International 1924
2000000037	Bermuda 1957	Clarke 1866
2000000038	Bern 1898	Bessel 1841
2000000039	Bern 1898 (Bern)	Bessel 1841
2000000040	Bern 1938	Bessel 1841
2000000041	Bessel 1841	Bessel 1841
2000000042	Bessel Modified	Bessel Modified
2000000043	Bessel Namibia	Bessel Namibia
2000000044	Bissau	International 1924
2000000045	Bogota	International 1924
2000000046	Bogota (Bogota)	International 1924
2000000047	Bukit Rimpah	Bessel 1841
2000000048	Camacupa	Clarke 1880 (RGS)
2000000049	Campo Inchauspe	International 1924
2000000050	Camp Area Astro	International 1924
2000000051	Canton Astro 1966	International 1924
2000000052	Cape	Clarke 1880 (Arc)
2000000053	Cape Canaveral	Clarke 1866
2000000054	Carthage	Clarke 1880 (IGN)
2000000055	Carthage (degrees)	Clarke 1880 (IGN)
2000000056	Carthage (Paris)	Clarke 1880 (IGN)
2000000057	CH 1903	Bessel 1841
2000000058	CH 1903+	Bessel 1841
2000000059	Chatham Island Astro 1971	International 1924
2000000060	Chos Malal 1914	International 1924
2000000061	Swiss Terrestrial Ref. Frame 1995	GRS 1980
2000000062	Chua	International 1924
2000000063	Clarke 1858	Clarke 1858

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000064	Clarke 1866	Clarke 1866
2000000065	Clarke 1866 (Michigan)	Clarke 1866 (Michigan)
2000000066	Clarke 1880	Clarke 1880
2000000067	Clarke 1880 (Arc)	Clarke 1880 (Arc)
2000000068	Clarke 1880 (Benoit)	Clarke 1880 (Benoit)
2000000069	Clarke 1880 (IGN)	Clarke 1880 (IGN)
2000000070	Clarke 1880 (RGS)	Clarke 1880 (RGS)
2000000071	Clarke 1880 (SGA)	Clarke 1880 (SGA)
2000000072	Conakry 1905	Clarke 1880 (IGN)
2000000073	Corrego Alegre	International 1924
2000000074	Cote d'Ivoire	Clarke 1880 (IGN)
2000000075	Dabola 1981	Clarke 1880 (RGS)
2000000076	Datum 73	International 1924
2000000077	Dealul Piscului 1933 (Romania)	International 1924
2000000078	Dealul Piscului 1970 (Romania)	Krasovsky 1940
2000000079	Deception Island	Clarke 1880 (RGS)
2000000080	Deir ez Zor	Clarke 1880 (IGN)
2000000081	Deutsche Hauptdreiecksnetz	Bessel 1841
2000000082	Dominica 1945	Clarke 1880 (RGS)
2000000083	DOS 1968	International 1924
2000000084	Astro DOS 71/4	International 1924
2000000085	Douala	Clarke 1880 (IGN)
2000000086	Easter Island 1967	International 1924
2000000087	European Datum 1950	International 1924
2000000088	European Datum 1950 (ED77)	International 1924
2000000089	European Datum 1987	International 1924
2000000090	Egypt 1907	Helmert 1906
2000000091	Estonia 1937	Bessel 1841
2000000092	Estonia 1992	GRS 1980
2000000093	European Terrestrial Ref. Frame 1989	WGS 1984
2000000094	European 1979	International 1924
2000000095	European Libyan Datum 1979	International 1924
2000000096	Everest 1830	Everest 1830
2000000097	Everest (Bangladesh)	Everest Adjustment 1937
2000000098	Everest (Definition 1962)	Everest (Definition 1962)
2000000099	Everest (Definition 1967)	Everest (Definition 1967)

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000100	Everest (Definition 1975)	Everest (Definition 1975)
2000000101	Everest (India and Nepal)	Everest (Definition 1962)
2000000102	Everest 1830 Modified	Everest 1830 Modified
2000000103	Everest Modified 1969	Everest Modified 1969
2000000104	Fahud	Clarke 1880 (RGS)
2000000105	Final Datum 1958	Clarke 1880 (RGS)
2000000106	Fischer 1960	Fischer 1960
2000000107	Fischer 1968	Fischer 1968
2000000108	Fischer Modified	Fischer Modified
2000000109	Fort Thomas 1955	Clarke 1880 (RGS)
2000000110	Gandajika 1970	International 1924
2000000111	Gan 1970	International 1924
2000000112	Garoua	Clarke 1880 (IGN)
2000000113	Geocentric Datum of Australia 1994	GRS 1980
2000000114	GEM 10C Gravity Potential Model	GEM 10C
2000000115	Greek Geodetic Ref. System 1987	GRS 1980
2000000116	Graciosa Base SW 1948	International 1924
2000000117	Greek	Bessel 1841
2000000118	Greek (Athens)	Bessel 1841
2000000119	Grenada 1953	Clarke 1880 (RGS)
2000000120	GRS 1967	GRS 1967
2000000121	GRS 1980	GRS 1980
2000000122	Guam 1963	Clarke 1866
2000000123	Gunung Segara	Bessel 1841
2000000124	GUX 1 Astro	International 1924
2000000125	Guyane Francaise	International 1924
2000000126	Hanoi 1972	Krasovsky 1940
2000000127	Hartebeesthoek 1994	WGS 1984
2000000128	Helmert 1906	Helmert 1906
2000000129	Herat North	International 1924
2000000130	Hermannskogel	Bessel 1841
2000000131	Hito XVIII 1963	International 1924
2000000132	Hjorsey 1955	International 1924
2000000133	Hong Kong 1963	International 1924
2000000134	Hong Kong 1980	International 1924
2000000135	Hough 1960	Hough 1960
2000000136	Hungarian Datum 1972	GRS 1967
2000000137	Hu Tzu Shan	International 1924

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000138	Indian 1954	Everest Adjustment 1937
2000000139	Indian 1960	Everest Adjustment 1937
2000000140	Indian 1975	Everest Adjustment 1937
2000000141	Indonesian National	Indonesian National
2000000142	Indonesian Datum 1974	Indonesian
2000000143	International 1927	International 1924
2000000144	International 1967	International 1967
2000000145	IRENET95	GRS 1980
2000000146	Israel	GRS 1980
2000000147	ISTS 061 Astro 1968	International 1924
2000000148	ISTS 073 Astro 1969	International 1924
2000000149	Jamaica 1875	Clarke 1880
2000000150	Jamaica 1969	Clarke 1866
2000000151	Japan Geodetic Datum 2000	GRS 1980
2000000152	Johnston Island 1961	International 1924
2000000153	Kalianpur 1880	Everest 1830
2000000154	Kalianpur 1937	Everest Adjustment 1937
2000000155	Kalianpur 1962	Everest (Definition 1962)
2000000156	Kalianpur 1975	Everest (Definition 1975)
2000000157	Kandawala	Everest Adjustment 1937
2000000158	Kerguelen Island 1949	International 1924
2000000159	Kertau	Everest 1830 Modified
2000000160	Kartastokoordinaattijarjestelma	International 1924
2000000161	Kuwait Oil Company	Clarke 1880 (RGS)
2000000162	Korean Datum 1985	Bessel 1841
2000000163	Korean Datum 1995	WGS 1984
2000000164	Krasovsky 1940	Krasovsky 1940
2000000165	Kuwait Utility	GRS 1980
2000000166	Kusaie Astro 1951	International 1924
2000000167	Lake	International 1924
2000000168	La Canoa	International 1924
2000000169	L.C. 5 Astro 1961	Clarke 1866
2000000170	Leigon	Clarke 1880 (RGS)
2000000171	Liberia 1964	Clarke 1880 (RGS)

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000172	Datum Lisboa Bessel	Bessel 1841
2000000173	Datum Lisboa Hayford	International 1924
2000000174	Lisbon	International 1924
2000000175	Lisbon (Lisbon)	International 1924
2000000176	LKS 1994	GRS 1980
2000000177	Locodjo 1965	Clarke 1880 (RGS)
2000000178	Loma Quintana	International 1924
2000000179	Lome	Clarke 1880 (IGN)
2000000180	Luzon 1911	Clarke 1866
2000000181	Madrid 1870 (Madrid Prime Merid.)	Struve 1860
2000000182	Madzansua	Clarke 1866
2000000183	Mahe 1971	Clarke 1880 (RGS)
2000000184	Majuro (Republic of Marshall Is.)	Clarke 1866
2000000185	Makassar	Bessel 1841
2000000186	Makassar (Jakarta)	Bessel 1841
2000000187	Malongo 1987	International 1924
2000000188	Manoca	Clarke 1880 (RGS)
2000000189	Massawa	Bessel 1841
2000000190	Merchich	Clarke 1880 (IGN)
2000000191	Merchich (degrees)	Clarke 1880 (IGN)
2000000192	Militar-Geographische Institut	Bessel 1841
2000000193	MGI (Ferro)	Bessel 1841
2000000194	Mhast	International 1924
2000000195	Midway Astro 1961	International 1924
2000000196	Minna	Clarke 1880 (RGS)
2000000197	Monte Mario	International 1924
2000000198	Monte Mario (Rome)	International 1924
2000000199	Montserrat Astro 1958	Clarke 1880 (RGS)
2000000200	Mount Dillon	Clarke 1858
2000000201	Moznet	WGS 1984
2000000202	M'poraloko	Clarke 1880 (IGN)
2000000203	North American Datum 1927	Clarke 1866
2000000204	NAD 1927 CGQ77	Clarke 1866
2000000205	NAD 1927 (1976)	Clarke 1866
2000000206	North American Datum 1983	GRS 1980
2000000207	NAD 1983 (Canadian Spatial Ref. System)	GRS 1980
2000000208	North American Datum 1983 (HARN)	GRS 1980
2000000209	NAD Michigan	Clarke 1866 (Michigan)

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000210	Nahrwan 1967	Clarke 1880 (RGS)
2000000211	Naparima 1955	International 1924
2000000212	Naparima 1972	International 1924
2000000213	Nord de Guerre (Paris)	Plessis 1817
2000000214	National Geodetic Network (Kuwait)	WGS 1984
2000000215	NGO 1948	Bessel Modified
2000000216	NGO 1948 (Oslo)	Bessel Modified
2000000217	Nord Sahara 1959	Clarke 1880 (RGS)
2000000218	NSWC 9Z-2	NWL 9D
2000000219	Nouvelle Triangulation Francaise (degrees)	Clarke 1880 (IGN)
2000000220	NTF (Paris) (grads)	Clarke 1880 (IGN)
2000000221	NWL 9D Transit Precise Ephemeris	NWL 9D
2000000222	New Zealand Geodetic Datum 1949	International 1924
2000000223	New Zealand Geodetic Datum 2000	GRS 1980
2000000224	Observatorio	Clarke 1866
2000000225	Observ. Meteorologico 1939	International 1924
2000000226	Old Hawaiian	Clarke 1866
2000000227	Oman	Clarke 1880 (RGS)
2000000228	OSGB 1936	Airy 1830
2000000229	OSGB 1970 (SN)	Airy 1830
2000000230	OSU 1986 Geoidal Model	OSU 86F
2000000231	OSU 1991 Geoidal Model	OSU 91A
2000000232	OS (SN) 1980	Airy 1830
2000000233	Padang 1884	Bessel 1841
2000000234	Padang 1884 (Jakarta)	Bessel 1841
2000000235	Palestine 1923	Clarke 1880 (Benoit)
2000000236	Pampa del Castillo	International 1924
2000000237	PDO Survey Datum 1993	Clarke 1880 (RGS)
2000000238	Pico de Las Nieves	International 1924
2000000239	Pitcairn Astro 1967	International 1924
2000000240	Plessis 1817	Plessis 1817
2000000241	Pohnpei (Fed. States of Micronesia)	Clarke 1866
2000000242	Point 58	Clarke 1880 (RGS)
2000000243	Pointe Noire	Clarke 1880 (IGN)
2000000244	Porto Santo 1936	International 1924
2000000245	POSGAR	GRS 1980
2000000246	Provisional South Amer. Datum 1956	International 1924
2000000247	Puerto Rico	Clarke 1866
2000000248	Pulkovo 1942	Krasovsky 1940

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000249	Pulkovo 1995	Krasovsky 1940
2000000250	Qatar 1974	International 1924
2000000251	Qatar 1948	Helmert 1906
2000000252	Qornoq	International 1924
2000000253	Rassadiran	International 1924
2000000254	REGVEN	GRS 1980
2000000255	Reunion	International 1924
2000000256	Reseau Geodesique Francais 1993	GRS 1980
2000000257	RT38	Bessel 1841
2000000258	RT38 (Stockholm)	Bessel 1841
2000000259	RT 1990	Bessel 1841
2000000260	S-42 Hungary	Krasovsky 1940
2000000261	South American Datum 1969	GRS 1967 Truncated
2000000262	Samboja	Bessel 1841
2000000263	American Samoa 1962	Clarke 1866
2000000264	Santo DOS 1965	International 1924
2000000265	Sao Braz	International 1924
2000000266	Sapper Hill 1943	International 1924
2000000267	Schwarzeck	Bessel Namibia
2000000268	Segora	Bessel 1841
2000000269	Selvagem Grande 1938	International 1924
2000000270	Serindung	Bessel 1841
2000000271	Sierra Leone 1924	War Office
2000000272	Sierra Leone 1960	Clarke 1880 (RGS)
2000000273	Sierra Leone 1968	Clarke 1880 (RGS)
2000000274	SIRGAS	GRS 1980
2000000275	South Yemen	Krasovsky 1940
2000000276	Authalic sphere	Sphere
2000000277	Authalic sphere (ARC/INFO)	Sphere ARC INFO
2000000278	Struve 1860	Struve 1860
2000000279	St. George Island (Alaska)	Clarke 1866
2000000280	St. Kitts 1955	Clarke 1880 (RGS)
2000000281	St. Lawrence Island (Alaska)	Clarke 1866
2000000282	St. Lucia 1955	Clarke 1880 (RGS)
2000000283	St. Paul Island (Alaska)	Clarke 1866
2000000284	St. Vincent 1945	Clarke 1880 (RGS)
2000000285	Sudan	Clarke 1880 (IGN)
2000000286	South Asia Singapore	Fischer Modified
2000000287	S-JTSK	Bessel 1841

Differenze nell'utilizzo dei dati geodetici e spaziali

Tabella 28. SRID con datum e ellissoide associati (Continua)

SRID	Nome datum	Ellissoide di riferimento
2000000288	S-JTSK (Ferro)	Bessel 1841
2000000289	Tananarive 1925	International 1924
2000000290	Tananarive 1925 (Paris)	International 1924
2000000291	Tern Island Astro 1961	International 1924
2000000292	Tete	Clarke 1866
2000000293	Timbalai 1948	Everest (Definition 1967)
2000000294	TM65	Airy Modified
2000000295	TM75	Airy Modified
2000000296	Tokyo	Bessel 1841
2000000297	Trinidad 1903	Clarke 1858
2000000298	Tristan Astro 1968	International 1924
2000000299	Trucial Coast 1948	Helmert 1906
2000000300	Viti Levu 1916	Clarke 1880 (RGS)
2000000301	Voirol 1875	Clarke 1880 (IGN)
2000000302	Voirol 1875 (degrees)	Clarke 1880 (IGN)
2000000303	Voirol 1875 (Paris)	Clarke 1880 (IGN)
2000000304	Voirol Unifie 1960	Clarke 1880 (RGS)
2000000305	Voirol Unifie 1960 (degrees)	Clarke 1880 (RGS)
2000000306	Voirol Unifie 1960 (Paris)	Clarke 1880 (RGS)
2000000307	Wake-Eniwetok 1960	Hough 1960
2000000308	Wake Island Astro 1952	International 1924
2000000309	Walbeck	Walbeck
2000000310	War Office	War Office
2000000311	WGS 1966	WGS 1966
2000000312	WGS 1972	WGS 1972
2000000313	WGS 1972 Transit Broadcast Ephemeris	WGS 1972
2000000314	Yacare	International 1924
2000000315	Yemen Nat'l Geodetic Network 1996	WGS 1984
2000000316	Yoff	Clarke 1880 (IGN)
2000000317	Zanderij	International 1924

Sferoidi geodetici

Uno sferoide (noto anche come ellissoide) è la parte di un sistema di coordinate geografiche che definisce la forma della superficie della Terra in una posizione specifica.

La definizione di un sistema di coordinate comprende la definizione di un ellissoide nella definizione SPHEROID che è parte della definizione DATUM, come mostrato nel seguente esempio:

Differenze nell'utilizzo dei dati geodetici e spaziali

```
GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",  
SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199432955]]
```

Per un elenco di sferoidi forniti da Spatial Extender e Geodetic Extender, consultare la sezione “Sistemi di coordinate supportati” a pagina 525. Queste informazioni sono contenute nella vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS. La colonna **DEFINITION** nella vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS contiene i valori delle colonne **Nome**, **Semiassse maggiore** e **Schiacciamento** nella tabella Sferoidi supportati.

Argomenti correlati:

- “Sistema di coordinate geografiche” a pagina 59

Riferimenti correlati:

- “Vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS” a pagina 283
- “ST_create_coordsys” a pagina 236

Differenze nell'utilizzo dei dati geodetici e spaziali

Parte 5. Materiale di riferimento

Capitolo 20. Procedure memorizzate

Questo capitolo contiene le informazioni di riferimento relative alle procedure memorizzate che è possibile utilizzare per impostare DB2 Spatial Extender e creare progetti che utilizzano dati spaziali. Durante l'impostazione di DB2 Spatial Extender o durante la creazione di progetti mediante il Centro di controllo DB2 o il processore riga comandi DB2, tali procedure vengono richiamate implicitamente. Ad esempio, scegliendo **OK** nella finestra DB2 Spatial Extender nel Centro di controllo DB2, viene richiamata la procedura memorizzata associata a quella finestra.

In alternativa, è possibile richiamare in modo esplicito le procedure memorizzate in un programma applicativo.

Per richiamare la maggior parte delle procedure memorizzate di DB2 Spatial Extender per un database, è necessario abilitare il database per le operazioni spaziali, richiamando la procedura `ST_enable_db` direttamente oppure utilizzando il Centro di controllo DB2. La sezione `ST_enable_dbe`, contenuto in questo capitolo, contiene le informazioni sul richiamo di questa procedura memorizzata.

Dopo aver abilitato un database per le operazioni spaziali, è possibile richiamare qualsiasi procedura memorizzata DB2 Spatial Extender, implicitamente o esplicitamente, per il database a cui si è collegati.

In questo capitolo sono descritte le seguenti procedure memorizzate di DB2 Spatial Extender:

- `GSE_export_sde`
- `GSE_import_sde`
- `ST_alter_coordsys`
- `ST_alter_srs`
- `ST_create_coordsys`
- `ST_create_srs`
- `ST_disable_autogeocoding`
- `ST_disable_db`
- `ST_drop_coordsys`
- `ST_drop_srs`
- `ST_enable_autogeocoding`
- `ST_enable_db`
- `ST_export_shape`
- `ST_import_shape`
- `ST_register_geocoder`
- `ST_register_spatial_column`
- `ST_remove_geocoding_setup`
- `ST_run_geocoding`
- `ST_setup_geocoding`
- `ST_unregister_geocoder`
- `ST_unregister_spatial_column`

Procedure memorizzate

Le implementazioni delle procedure memorizzate sono archiviate nella libreria db2gse sul server DB2 Spatial Extender.

GSE_export_sde

Utilizzare questa procedura memorizzata per esportare una colonna spaziale e la tabella associata in un file di trasferimento SDE.

Limitazioni:

- La tabella o la vista possono contenere una sola colonna spaziale.
- E' necessario che la colonna spaziale sia registrata.
- Non è possibile utilizzare file SDE esistenti.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM. Inoltre, dovrà disporre del privilegio SELECT per la tabella che si desidera esportare.

Sintassi:

```
▶▶ db2gse.GSE_export_sde ( schema_tabella | nullo , nome_tabella , nome_colonna , nome_file , clausola_where | nullo )
```

Descrizioni dei parametri:

schema_tabella

Indica il nome dello schema a cui appartiene la tabella indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della funzione che si desidera esportare. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica la colonna che si desidera esportare. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_file

Indica il nome del file di trasferimento SDE in cui verranno esportate la colonna spaziale indicata e la tabella ad essa associata. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

clausola_where

Indica il testo della clausola SQL WHERE, che definisce una limitazione nella serie di record da esportare. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, nella clausola WHERE non verrà definita alcuna limitazione.

Se il parametro viene specificato, il valore può fare riferimento a qualsiasi colonna di attributi nella tabella che si sta esportando.

Il tipo di dati utilizzato con questo parametro è VARCHAR(1024).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata GSE_export_sde. Per esportare i dati dalla tabella CUSTOMERS nei file SDE, viene utilizzato il comando DB2 CALL:

```
call db2gse.GSE_export_sde(NULL,'CUSTOMERS','LOCATION',
'/tmp/file_sde_di_esportazione', NULL,?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "GSE_import_sde" a pagina 228

GSE_import_sde

Utilizzare questa procedura memorizzata per importare un file di trasferimento SDE in un database abilitato per operazioni spaziali. La procedura memorizzata può funzionare in base alle seguenti modalità:

- Se il file di trasferimento SDE è destinato a una tabella esistente che dispone di una colonna spaziale registrata, DB2 Spatial Extender inserisce nella tabella i dati del file.
- In caso contrario, DB2 Spatial Extender crea una tabella con una colonna spaziale, registra la colonna e inserisce nella colonna spaziale e nelle altre colonne della tabella i dati del file.

Il sistema di riferimento spaziale specificato nel file di trasferimento SDE viene confrontato con i sistemi di riferimento spaziali registrati in DB2 Spatial Extender. Se il sistema specificato corrisponde a un sistema registrato, tutti i valori dei dati di trasferimento, una volta caricati, vengono modificati come indicato dal sistema registrato. Se il sistema specificato non corrisponde ad alcun sistema registrato, DB2 Spatial Extender crea un nuovo sistema di riferimento spaziale in cui indicare le modifiche.

Autorizzazione:

Quando si importano i dati in una tabella esistente, l'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verranno importati i dati
- Privilegio CONTROL per questa tabella

L'ID utente utilizzato per richiamare la procedura memorizzata per la creazione della tabella in cui importare i dati, deve disporre dell'autorizzazione SYSADM o DBADM per il database conterrà la tabella.

Sintassi:

```
▶▶ db2gse.GSE_import_sde—(—schema_tabella—,—nome_tabella—,——————▶
                        |nulla—)
▶—nome_colonna—,—nome_file—,—ambito_commit—)—————▶
                        |nulla—)
```

Descrizioni dei parametri:

schema_tabella

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella in cui verranno caricati i dati di trasferimento SDE. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica il nome della colonna registrate in cui verranno caricati i dati spaziali del file di trasferimento SDE. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_file

Indica il nome del file di trasferimento SDE che si desidera importare. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

ambito_commit

Indica il numero di record da importare prima di eseguire un COMMIT. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero) e nessun record viene sottoposto a commit.

Il tipo di dati utilizzato con questo parametro è INTEGER.

Parametri di output:*codice_msg*

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata GSE_import_sde. Per importare il file SDE tmp/customerSDE nella tabella CUSTOMERS, viene utilizzato il comando DB2 CALL. Il comando CALL indica di eseguire un COMMIT dopo ogni 5 record importati:

GSE_import_sde

```
call db2gse.GSE_import_sde(NULL,'CUSTOMERS','LOCATION',  
    '/tmp/customerSde', 5, ?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "GSE_export_sde" a pagina 226

ST_alter_coordsys

Utilizzare questa procedura memorizzata per aggiornare la definizione di un sistema di coordinate nel database. Durante l'esecuzione della procedura, le informazioni relative al sistema di coordinate vengono aggiornate nella vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS.

Attenzione: Se la procedura memorizzata viene utilizzata per modificare la definizione di un sistema di coordinate su cui si basa un sistema di riferimento spaziale associato a dati spaziali esistenti, tali dati potrebbero essere modificati durante l'utilizzo della procedura. In questo caso, assicurarsi che i dati spaziali siano ancora corretti e validi.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM.

Sintassi:

```
►► db2gse.ST_alter_coordsys( (—nome_sistcoord—, —definizione—, —————→  
    [nullo] )  
► [organizzazione] , [ID_sistcoord_organizzazione] , [descrizione] →  
    [nullo] [nullo] [nullo]  
►) —————→
```

Descrizioni dei parametri:

nome_sistcoord

Identifica il sistema di coordinate. Il valore del parametro non può essere nullo.

Il valore *nome_sistcoord* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

definizione

Definisce il sistema di coordinate. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, la definizione del sistema non viene modificata.

Il tipo di dati utilizzato con questo parametro è VARCHAR(2048).

organizzazione

Indica il nome dell'organizzazione che definisce il sistema di coordinate e ne

fornisce una definizione; ad esempio "European Petroleum Survey Group (EPSG)". Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo.

Se il parametro è nullo, l'organizzazione del sistema di coordinate non viene modificata. Se il parametro non è nullo, il parametro *id_sistcoord_organizzazione* non potrà essere nullo; in questo caso, la combinazione tra i parametri *organizzazione* e *id_sistcoord_organizzazione* identificherà univocamente il sistema di coordinate.

Il tipo di dati utilizzato con questo parametro è VARCHAR(128).

id_sistcoord_organizzazione

Specifica un identificativo numerico assegnato a questo sistema di coordinate dall'entità identificata dal parametro *organizzazione*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo.

Se il parametro è nullo, il parametro *organizzazione* dovrà essere nullo; in questo caso, l'identificativo del sistema di coordinate dell'organizzazione non verrà modificato. Se il parametro non è nullo, il parametro *organizzazione* non potrà essere nullo; in questo caso la combinazione tra i parametri *organizzazione* e *id_sistcoord_organizzazione* identificherà univocamente il sistema di coordinate.

Il tipo di dati utilizzato con questo parametro è INTEGER.

descrizione

Descrive il sistema di coordinate, descrivendo la relativa applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, la descrizione del sistema di coordinate non viene modificata.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_alter_coordsys. Per aggiornare il sistema di coordinate NORTH_AMERICAN_TEST, viene utilizzato il comando DB2 CALL. Il comando CALL assegna il valore 1002 al parametro *id_sistcoord*:

```
call db2gse.ST_alter_coordsys('NORTH_AMERICAN_TEST',NULL,NULL,1002,NULL,?,?)
```

ST_alter_coordsys

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_create_coordsys" a pagina 236
- "ST_drop_coordsys" a pagina 248

ST_alter_srs

Utilizzare questa procedura memorizzata per aggiornare la definizione di un sistema di riferimento spaziale nel database. Durante l'esecuzione di questa procedura memorizzata, le informazioni relative al sistema di riferimento spaziale vengono aggiornate nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS.

DB2 Spatial Extender memorizza internamente i valori delle coordinate come numeri interi positivi. In tal modo, durante le operazioni di calcolo, si riduce la possibilità di commettere errori di arrotondamento (che dipendono soprattutto dal valore reale delle operazioni a virgola mobile) e si migliorano significativamente le prestazioni delle operazioni spaziali.

Limitazione: Non è possibile modificare un sistema di riferimento spaziale se esiste una colonna spaziale registrata che utilizza il sistema.

Attenzione: Se la procedura memorizzata viene utilizzata per modificare i parametri offset, scala o *nome_sistcoord* del sistema di riferimento spaziale, ed esistono dati spaziali associati a quel sistema, tali dati potrebbero essere modificati durante l'utilizzo della procedura. In questo caso, assicurarsi che i dati spaziali siano ancora corretti e validi.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM.

Sintassi:

```
▶▶ db2gse.ST_alter_srs ( ( nome_srs , srs_id , offset_x ,
                        [ nullo ] [ nullo ] ,
▶ scala_x , offset_y , scala_y , offset_z ,
  [ nullo ] [ nullo ] [ nullo ] [ nullo ] ,
▶ scala_z , offset_m , scala_m , nome_sistcoord ,
  [ nullo ] [ nullo ] [ nullo ] [ nullo ] ,
▶ descrizione ) ) ▶▶
```

Descrizioni dei parametri:

nome_srs

Identifica il sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

srs_id

Identifica il sistema di riferimento spaziale. Questo identificativo viene utilizzato come parametro di input in diverse funzioni spaziali. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, l'identificativo numerico del sistema di riferimento spaziale non viene modificato.

Il tipo di dati utilizzato con questo parametro è INTEGER.

offset_x

Indica l'offset per tutte le coordinate X delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_x* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender. *WKT* indica well-known text e *WKB* well-known binary.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_x

Indica il fattore di scala per tutte le coordinate X delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_x* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_y

Indica l'offset per tutte le coordinate Y delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_y* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_y

Indica il fattore di scala per tutte le coordinate Y delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_y* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender. Il fattore di scala deve essere uguale al fattore *scala_x*.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_z

Indica l'offset per tutte le coordinate Z delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

L'offset viene sottratto prima dell'applicazione del fattore di scala *z_scale* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_z

Indica il fattore di scala per tutte le coordinate Z delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_z* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_m

Indica l'offset per tutte le coordinate M delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_m* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_m

Indica il fattore di scala per tutte le coordinate M delle forme geometriche rappresentate nel sistema di riferimento spaziale. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il valore del parametro nella definizione del sistema di riferimento spaziale non viene modificato.

Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_m* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, forma) in rappresentazioni interne di DB2 Spatial Extender.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

nome_sistcoord

Identifica il tipo di sistema di coordinate su cui si basa il sistema di riferimento

spaziale. E' necessario che il sistema di coordinate sia elencato nella vista ST_COORDINATE_SYSTEMS. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il sistema di coordinate utilizzato per il sistema di riferimento spaziale non viene modificato.

Il valore *nome_sistcoord* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

descrizione

Descrive il sistema di riferimento spaziale, descrivendo la relativa applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, la descrizione del sistema di riferimento spaziale non viene modificata.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_alter_srs. Per modificare il valore del parametro *descrizione* del sistema di riferimento spaziale SRSDEMO, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_alter_srs('SRSDEMO',NULL,NULL,NULL,NULL,NULL,NULL,NULL,
NULL,NULL,'SRS for GSE Demo Program: offices table',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_drop_srs" a pagina 249
- "ST_create_srs" a pagina 238

Se il parametro è nullo, il parametro *organizzazione* dovrà essere nullo. Se il parametro non è nullo, il parametro *organizzazione* non potrà essere nullo; in questo caso la combinazione tra i parametri *organizzazione* e *id_sistcoord_organizzazione* identificherà univocamente il sistema di coordinate.

Il tipo di dati utilizzato con questo parametro è INTEGER.

descrizione

Descrive il sistema di coordinate, descrivendo la relativa applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà registrata alcuna informazione sul sistema di coordinate.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_create_coordsys. Per creare il sistema di coordinate con i seguenti parametri, procedere come segue:

- Parametro *nome_sistcoord*: NORTH_AMERICAN_TEST
- Parametro *definizione*:
 GEOGCS["GCS_North_American_1983",
 DATUM["D_North_American_1983",
 SPHEROID["GRS_1980",6378137.0,298.257222101]],
 PRIMEM["Greenwich",0.0],
 UNIT["Degree",0.0174532925199433]]
- Parametro *organizzazione*: EPSG
- Parametro *id_sistcoord_organizzazione*: 1001
- Parametro *descrizione*: Test Coordinate Systems

```
call db2gse.ST_create_coordsys('NORTH_AMERICAN_TEST',
'GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",
SPHEROID["GRS_1980",6378137.0,298.257222101]],
PRIMEM["Greenwich",0.0],UNIT["Degree",
0.0174532925199433]]','EPSG',1001,'Test Coordinate Systems',?,?)
```

ST_create_coordsys

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_drop_srs" a pagina 249
- "ST_alter_srs" a pagina 232

ST_create_srs

Utilizzare questa procedura memorizzata per creare un sistema di riferimento spaziale. Il sistema di riferimento spaziale viene definito dal sistema di coordinate, dal livello di precisione e dall'estensione delle coordinate rappresentate nel sistema. Le estensioni sono i valori di coordinate minimi e massimi delle coordinate X, Y, Z e M.

DB2 Spatial Extender memorizza internamente i valori delle coordinate come numeri interi positivi. In tal modo, durante le operazioni di calcolo, si riduce la possibilità di commettere errori di arrotondamento (che dipendono soprattutto dal valore reale delle operazioni a virgola mobile) e si migliorano significativamente le prestazioni delle operazioni spaziali.

La procedura memorizzata dispone di due varianti:

- La prima variante utilizza i fattori di conversione (valori di scarto e fattori di scala) come parametri di input.
- La seconda variante utilizza le estensioni e il valore di precisione come parametri di input e calcola i fattori di conversione internamente.

Questa procedura memorizzata sostituisce la procedura `db2gse.gse_enable_sref`.

Autorizzazione:

Nessuna.

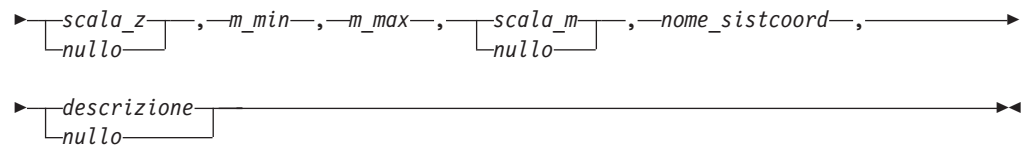
Sintassi:

Con i fattori di conversione (variante 1):

```
►► db2gse.ST_create_srs(—nome_srs—, —srs_id—, —offset_x—, —scala_x—,
                        └─nullo─┘
►, —offset_y—, —scala_y—, —offset_z—, —scala_z—,
  └─nullo─┘ └─nullo─┘ └─nullo─┘ └─nullo─┘
►, —offset_m—, —scala_m—, —nome_sistcoord—, —descrizione—)
  └─nullo─┘ └─nullo─┘ └─nullo─┘
```

Con l'estensione massima (variante 2):

```
►► db2gse.ST_create_srs(—nome_srs—, —srs_id—, —x_min—, —x_max—,
                        —scala_x—, —, —y_min—, —y_max—, —scala_y—, —z_min—, —z_max—,
                        └─nullo─┘
```

Descrizioni dei parametri:

Con i fattori di conversione (variante 1):

nome_srs

Identifica il sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

srs_id

Identifica il sistema di riferimento spaziale. Questo identificativo numerico viene utilizzato come parametro di input in diverse funzioni spaziali. Il valore del parametro non può essere nullo.

Per un sistema di riferimento spaziale geodetico, il valore *srs_id* deve essere compreso tra 2000000318 e 2000001000. DB2 Geodetic Extender fornisce sistemi di riferimento spaziali geodetici con valori di *srs_id* tra 2000000000 e 2000000317.

Il tipo di dati utilizzato con questo parametro è INTEGER.

offset_x

Indica l'offset per tutte le coordinate X delle forme geometriche rappresentate nel sistema di riferimento spaziale. L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_x* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. WKT indica well-known text e WKB well-known binary. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero).

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_x

Indica il fattore di scala per tutte le coordinate X delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_x* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Se il valore *offset_x* non viene indicato, viene utilizzato il valore *offset_x* predefinito pari a 0. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_y

Indica l'offset per tutte le coordinate Y delle forme geometriche rappresentate nel sistema di riferimento spaziale. L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_y* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero).

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_y

Indica il fattore di scala per tutte le coordinate Y delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_y* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Se il valore *offset_y* non viene indicato, viene utilizzato il valore *offset_y* predefinito pari a 0. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore del parametro *scala_x*. Se viene indicato un valore non nullo, è necessario che questo valore corrisponda al valore del parametro *scala_x*.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_z

Indica l'offset per tutte le coordinate Z delle forme geometriche rappresentate nel sistema di riferimento spaziale. esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero).

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_z

Indica il fattore di scala per tutte le coordinate Z delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_z* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Se il valore *offset_z* non viene indicato, viene utilizzato il valore *offset_z* predefinito pari a 0. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 1.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

offset_m

Indica l'offset per tutte le coordinate M delle forme geometriche rappresentate nel sistema di riferimento spaziale. L'offset viene sottratto prima dell'applicazione del fattore di scala *scala_m* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero).

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_m

Indica il fattore di scala per tutte le coordinate M delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_m* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Se il valore *offset_m* non viene indicato, viene utilizzato il valore *offset_m* predefinito pari a 0. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 1.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

nome_sistcoord

Identifica il tipo di sistema di coordinate su cui si basa il sistema di riferimento spaziale. E' necessario che il sistema di coordinate sia elencato nella vista ST_COORDINATE_SYSTEMS. Il valore del parametro non può essere nullo.

Il valore *nome_sistcoord* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

descrizione

Descrive il sistema di riferimento spaziale illustrando lo scopo dell'applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà registrata alcuna informazione.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Con l'estensione massima (variante 2):*nome_srs*

Identifica il sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

srs_id

Identifica il sistema di riferimento spaziale. Questo identificativo numerico viene utilizzato come parametro di input in diverse funzioni spaziali. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è INTEGER.

x_min

Indica il valore minimo possibile della coordinata X per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

x_max

Indica il valore massimo possibile della coordinata X per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore contenuto nella vista DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS potrebbe essere maggiore del valore specificato, in base al valore di *scala_x*. Il valore nella vista è il valore corretto.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_x

Indica il fattore di scala per tutte le coordinate X delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_x* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB,

shape) in rappresentazioni interne di DB2 Spatial Extender. Il calcolo dell'offset *offset_x* si basa sul valore di *x_min*. Il valore del parametro non può essere nullo.

Se i parametri *scala_x* e *scala_y* vengono entrambi specificati, i valori dovranno corrispondere.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

y_min

Indica il valore minimo possibile della coordinata Y per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

y_max

Indica il valore massimo possibile della coordinata Y per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore contenuto nella vista DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS potrebbe essere maggiore del valore specificato, in base al valore di *scala_y*. Il valore nella vista è il valore corretto.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_y

Indica il fattore di scala per tutte le coordinate Y delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_y* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Il calcolo dell'offset *offset_y* si basa sul valore di *y_min*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore del parametro *scala_x*. Se i parametri *scala_y* e *scala_x* vengono entrambi specificati, i valori dovranno corrispondere.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

z_min

Indica il valore minimo possibile della coordinata Z per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

z_max

Indica il valore massimo possibile della coordinata Z per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore contenuto nella vista DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS potrebbe essere maggiore del valore specificato, in base al valore di *scala_z*. Il valore nella vista è il valore corretto.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_z

Indica il fattore di scala per tutte le coordinate Z delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_z* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB,

shape) in rappresentazioni interne di DB2 Spatial Extender. Il calcolo dell'offset *offset_z* si basa sul valore di *z_min*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 1.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

m_min

Indica il valore minimo possibile della coordinata M per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

m_max

Indica il valore massimo possibile della coordinata M per tutte le forme geometriche che utilizzano questo sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore contenuto nella vista DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS potrebbe essere maggiore del valore specificato, in base al valore di *scala_m*. Il valore nella vista è il valore corretto.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

scala_m

Indica il fattore di scala per tutte le coordinate M delle forme geometriche rappresentate nel sistema di riferimento spaziale. Il fattore di scala viene applicato (moltiplicazione) dopo la sottrazione dell'offset *offset_m* durante la conversione delle forme geometriche da rappresentazioni esterne (WKT, WKB, shape) in rappresentazioni interne di DB2 Spatial Extender. Il calcolo dell'offset *offset_m* si basa sul valore di *m_min*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 1.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

nome_sistcoord

Identifica il tipo di sistema di coordinate su cui si basa il sistema di riferimento spaziale. E' necessario che il sistema di coordinate sia elencato nella vista ST_COORDINATE_SYSTEMS. Il valore del parametro non può essere nullo.

Il valore *nome_sistcoord* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

descrizione

Descrive il sistema di riferimento spaziale illustrando lo scopo dell'applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà registrata alcuna informazione.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza,

ST_create_srs

significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_create_srs. Per creare il sistema di riferimento spaziale SRSDemo con i seguenti parametri, viene utilizzato il comando DB2 CALL:

- *srs_id*: 1000000
- *offset_x*: -180
- *scala_x*: 1000000
- *offset_y*: -90
- *scala_y*: 1000000

```
call db2gse.ST_create_srs('SRSDemo',1000000,  
                          -180,1000000, -90, 1000000,  
                          0, 1, 0, 1,'NORTH_AMERICAN',  
                          'SRS for GSE Demo Program: customer table',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Argomenti correlati:

- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- "Creazione di un sistema di riferimento spaziale" a pagina 75

ST_disable_autogeocoding

Utilizzare questa procedura memorizzata per indicare a DB2 Spatial Extender di interrompere la sincronizzazione di una colonna sottoposta a geocoding con le colonne associate. Come input del geocoder viene utilizzata una *colonna di geocoding*.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_disable_autogc.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella in cui sono definiti i trigger da eliminare

- Privilegio CONTROL per questa tabella
- Privilegi ALTER e UPDATE per questa tabella

Nota: Per i privilegi CONTROL e ALTER, è necessario disporre di autorizzazione DROPIN per lo schema DB2GSE.

Sintassi:

```
▶—db2gse.ST_disable_autogeocoding—(—schema_tabella—,—nome_tabella—,—▶
└—nulla—┘
▶—nome_colonna—)—————▶
```

Descrizioni dei parametri:

schema_tabella

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome della tabella che contiene la definizione dei trigger da eliminare. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Il nome della colonna sottoposta a geocoding gestita dai trigger che si desidera eliminare. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura, si è verificata una o condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

ST_disable_autogeocoding

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_disable_autogeocoding. Per disabilitare la funzione di geocoding automatica nella colonna LOCATION della tabella CUSTOMERS, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_disable_autogeocoding(NULL, 'CUSTOMERS', 'LOCATION', ?, ?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_enable_autogeocoding" a pagina 250
- "ST_setup_geocoding" a pagina 276

ST_disable_db

Utilizzare questa procedura memorizzata per eliminare le risorse che consentono a DB2 Spatial Extender di memorizzare i dati spaziali e supportare le operazioni eseguite su questi dati.

La procedura memorizzata è di aiuto nella risoluzione dei problemi che possono verificarsi dopo l'abilitazione del database per le operazioni spaziali. Ad esempio, è possibile abilitare un database per le operazioni spaziali e in seguito decidere di utilizzare un database diverso con DB2 Spatial Extender. Se non è stata definita alcuna colonna spaziale o se non sono ancora stati importati dati spaziali, è possibile richiamare la procedura memorizzata per eliminare tutte le risorse spaziali dal primo database. Dal momento che le colonne spaziali e le definizioni dei tipi sono dipendenti tra loro, non sarà possibile eliminare le definizioni senza aver prima eliminato le colonne relative ai tipi a cui le definizioni si riferiscono. Se le colonne spaziali sono già state definite e si desidera comunque disabilitare il database per le operazioni spaziali, è necessario specificare un valore diverso da 0 (zero) per il parametro *force* in modo da eliminare tutte le risorse spaziali nel database.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_disable_db.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database in cui si desidera eliminare le risorse DB2 Spatial Extender.

Sintassi:

►► db2gse.ST_disable_db((*force*))

Descrizioni dei parametri:

force

Indicare se si desidera disabilitare un database per le operazioni spaziali, anche se il database contiene oggetti dipendenti da tipi o da funzioni spaziali. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se viene specificato un valore diverso da 0 (zero) o che non sia nullo per il parametro *force*, il database viene disabilitato, e tutte le risorse di DB2 Spatial Extender saranno eliminate (se possibile). Se viene indicato 0 (zero) o un valore nullo, se esistono oggetti del database dipendenti da tipi o funzioni spaziali, il database non sarà disabilitato. Gli oggetti del database che potrebbero avere tali dipendenze sono le tabelle, le viste, i limiti, i trigger, le colonne generate, i metodi, le funzioni, le procedure e altri tipi di dati (tipi secondari o tipi strutturati con un attributo spaziale).

Il tipo di dati utilizzato con questo parametro è SMALLINT.

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura, si è verificata una o condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_disable_db. Per disabilitare il database per le operazioni spaziali, viene utilizzato il comando DB2 CALL con un parametro *force* di valore 1:

```
call db2gse.ST_disable_db(1,?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_alter_coordsys" a pagina 230
- "ST_create_coordsys" a pagina 236

ST_drop_coordsys

Utilizzare questa procedura memorizzata per eliminare le informazioni relative a un sistema di coordinate dal database. Durante l'esecuzione della procedura, le informazioni relative al sistema di coordinate vengono eliminate dalla vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS.

Limitazione: Non è possibile eliminare un sistema di coordinate a cui fa riferimento un sistema di riferimento spaziale.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM.

Sintassi:

```
►►—db2gse.ST_drop_coordsys—(—nome_sistcoord—)—————►►
```

Descrizioni dei parametri:

nome_sistcoord

Identifica il sistema di coordinate. Il valore del parametro non può essere nullo.

Il valore *nome_sistcoord* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_drop_coordsys. Per eliminare il sistema di coordinate NORTH_AMERICAN_TEST dal database, viene utilizzato il comando DB2 CALL.

```
call db2gse.ST_drop_coordsys('NORTH_AMERICAN_TEST',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

ST_drop_srs

Utilizzare questa procedura memorizzata per eliminare un sistema di riferimento spaziale. Durante l'esecuzione di questa procedura memorizzata, le informazioni relative al sistema di riferimento spaziale vengono eliminate dalla vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS.

Limitazione: Non è possibile eliminare un sistema di riferimento spaziale se la colonna che utilizza il sistema è stata registrata.

Importante: Prestare molta attenzione durante l'utilizzo di questa procedura memorizzata. Se la procedura memorizzata viene utilizzata per eliminare un sistema di riferimento spaziale, non sarà più possibile eseguire operazioni spaziali con i dati associati al sistema.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_disable_sref.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM.

Sintassi:

```
▶▶—db2gse.ST_drop_srs—(—nome_srs—)—————▶▶
```

Descrizioni dei parametri:

nome_srs

Identifica il sistema di riferimento spaziale. Il valore del parametro non può essere nullo.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito

ST_drop_srs

dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_drop_srs. Per eliminare il sistema di riferimento spaziale SRSDEMO, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_drop_srs('SRSDEMO',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_create_srs" a pagina 238
- "ST_alter_srs" a pagina 232

ST_enable_autogeocoding

Utilizzare questa procedura memorizzata per indicare a DB2 Spatial Extender di sincronizzare una colonna sottoposta a geocoding con le colonne associate. Come input del geocoder viene utilizzata una *colonna di geocoding*. Ogni qual volta vengono inseriti nuovi valori, o vengono aggiornati i valori esistenti nelle colonne di geocoding, i trigger vengono attivati. I trigger richiamano il geocoder associato affinché esegua il geocoding dei nuovi valori e inserisca i risultati nell'apposita colonna.

Limitazione: E' possibile abilitare la funzione di geocoding automatica solo per le tabelle in cui è possibile creare i trigger INSERT e UPDATE. Di conseguenza, non è possibile abilitare la funzione di geocoding automatica per le viste o i soprannomi.

Prerequisiti: Prima di abilitare la funzione di geocoding automatica, è necessario impostare la funzione richiamando la procedura memorizzata ST_setup_geocoding. Durante l'impostazione della funzione di geocoding vengono specificati il geocoder e i valori dei parametri di geocoding. Inoltre, vengono identificate le colonne che dovranno essere sincronizzate con le colonne sottoposte a geocoding.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_enable_autogc.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella in cui sono definiti i trigger creati da questa procedura memorizzata
- Privilegio CONTROL per la tabella
- Privilegio ALTER per la tabella

ST_enable_autogeocoding

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_enable_autogeocoding. Per abilitare la funzione di geocoding automatica nella colonna LOCATION della tabella CUSTOMERS, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_enable_autogeocoding(NULL,'CUSTOMERS','LOCATION',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_setup_geocoding" a pagina 276

ST_enable_db

Utilizzare questa procedura memorizzata per fornire al database risorse necessarie per memorizzare dati spaziali e supportare le operazioni spaziali. Queste risorse includono tipi di dati e di indici spaziali, viste di catalogo, funzioni e procedure memorizzate.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_enable_db.

Autorizzazione:

L'ID utente con il quale la procedura memorizzata viene richiamata deve disporre dell'autorizzazione SYSADM o DBADM per il database che si desidera abilitare.

Sintassi:

```
►► db2gse.ST_enable_db ( parametri_creazione_tabella ) ►►  
                          └─ nullo ─┘
```

Descrizioni dei parametri:

parametri_creazione_tabella

Indica le opzioni da aggiungere alle istruzioni CREATE TABLE per le tabelle del catalogo di DB2 Spatial Extender. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, alle istruzioni CREATE TABLE non verrà aggiunta alcuna opzione.

Per indicare le opzioni desiderate, utilizzare la sintassi dell'istruzione DB2 CREATE TABLE. Ad esempio, per indicare un tablespace in cui creare le tabelle, utilizzare:

```
IN tsName INDEX IN indexTsName
```

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Il seguente esempio mostra come utilizzare la CLI (Call Level Interface) per richiamare la procedura memorizzata ST_enable_db:

```
SQLHANDLE henv;
SQLHANDLE hdbc;
SQLHANDLE hstmt;
SQLCHAR uid[MAX_UID_LENGTH + 1];
SQLCHAR pwd[MAX_PWD_LENGTH + 1];
SQLINTEGER ind[3];
SQLINTEGER codice_msg = 0;
char testo_msg[1024] = "";
SQLRETURN rc;
char *parametri_creazione_tabella = NULL;

/* Assegnare handle di ambiente */
rc = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv);

/* Assegnare handle di database */
rc = SQLAllocHandle(SQL_HANDLE_DBC, henv, &hdbc);

/* Stabilire una connessione al database "testdb" */
rc = SQLConnect(hdbc, (SQLCHAR *)"testdb", SQL_NTS, (SQLCHAR *)uid, SQL_NTS,
               (SQLCHAR *)pwd, SQL_NTS);

/* Assegnare handle di istruzione */
rc = SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);

/* Associare l'istruzione SQL per il richiamo della
procedura memorizzata ST_enable_db */
/* all'handle di istruzione ed inviare l'istruzione a DBMS. */
rc = SQLPrepare(hstmt, "call db2gse!ST_enable_db(?,?,?)", SQL_NTS);

/* Eseguire il bind del primo indicatore di parametro
nell'istruzione di richiamo, il parametro di */
/* input per i parametri di creazione della tabella, alla variabile */
/* parametri_creazione_tabella. */
ind[0] = SQL_NULL_DATA;
rc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT, SQL_C_CHAR,
                    SQL_VARCHAR, 255, 0, parametri_creazione_tabella, 256, &ind[0]);
```

ST_enable_db

```
/* Eseguire il bind del secondo indicatore di parametro
nell'istruzione di richiamo, */
/* il parametro di output per il codice messaggi restituite
alla variabile codice_msg. */
ind[1] = 0;
rc = SQLBindParameter(hstmt, 2, SQL_PARAM_OUTPUT, SQL_C_LONG,
                      SQL_INTEGER, 0, 0, &codice_msg, 4, &ind[1]);

/* Eseguire il bind del terzo indicatore di parametro
nell'istruzione di richiamo SQL, */
/* il parametro di output del testo del messaggio restituito,
alla variabile testo_msg. */
ind[2] = 0;
rc = SQLBindParameter(hstmt, 3, SQL_PARAM_OUTPUT, SQL_C_CHAR,
                      SQL_VARCHAR, (sizeof(testo_msg)-1), 0, testo_msg,
                      sizeof(testo_msg), &ind[2]);

rc = SQLExecute(hstmt);
```

Riferimenti correlati:

- “ST_disable_db” a pagina 246

ST_export_shape

Utilizzare questa procedura memorizzata per esportare una colonna spaziale e la tabella associata in un file shape.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_export_shape.

Autorizzazione:

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre dei privilegi necessari per eseguire l'istruzione SELECT da cui esportare i dati.

La procedura memorizzata, che viene eseguita come un processo appartenente al proprietario dell'istanza DB2, deve disporre dei privilegi necessari per creare o modificare i file shape sulla macchina server.

Sintassi:

```
db2gse.ST_export_shape(—nome_file—, —indicatore_accodamento—, —
                        |—nullo—|
—nome_colonna_output—, —istruzione_selezione—, —file_messaggi—)
|—nullo—| |—nullo—|
```

Descrizioni dei parametri:

nome_file

Indica il nome completo del file shape in cui esportare i dati specificati. Il valore del parametro non può essere nullo.

E' possibile utilizzare la procedura memorizzata ST_export_shape per esportare i dati in un nuovo file oppure accodare i dati a un file esistente:

- Se si desidera esportare i dati in un nuovo file, è possibile specificare l'estensione .shp o .SHP. Scegliendo una delle due estensioni, DB2 Spatial Extender crea il file con il valore *nome_file* specificato. Se non viene

specificata alcuna estensione, DB2 Spatial Extender crea un file avente lo stesso nome del valore *nome_file* specificato e con estensione .shp.

- Se si desidera aggiungere i dati esportati ad un file esistente, DB2 Spatial Extender ricerca la corrispondenza esatta del nome specificato per il parametro *nome_file*. Se non viene individuata alcuna corrispondenza, DB2 Spatial Extender ricercherà un file con estensione .shp, quindi uno con estensione .SHP.

Se il valore del parametro *indicatore_accodamento* indica che non si desidera aggiungere i dati ad un file esistente, ma che il file indicato con il parametro *nome_file* esiste già, DB2 Spatial Extender restituisce un errore e non sovrascrive il file.

La sezione “Note di utilizzo” a pagina 256 contiene un elenco dei file presenti sulla macchina server. La procedura memorizzata, che viene eseguita come un processo appartenente al proprietario dell’istanza DB2, deve disporre dei privilegi necessari per creare o modificare i file sulla macchina server.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

indicatore_accodamento

Indica se aggiungere i dati esportati ad un file shape esistente. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Per indicare se aggiungere i dati ad un file shape esistente, procedere come segue:

- Se si desidera aggiungere i dati ad un file shape esistente, indicare un qualsiasi valore non nullo e diverso da 0 (zero). In questo caso, la struttura di file dovrà corrispondere ai dati esportati; in caso contrario viene restituito un errore.
- Se si desidera esportare i file in un nuovo file, indicare un valore nullo oppure 0 (zero). In questo caso, DB2 Spatial Extender non sovrascriverà i file esistenti.

Il tipo di dati utilizzato con questo parametro è SMALLINT.

nome_colonna_output

Indica uno o più nomi di colonna (separati da virgole) da utilizzare per le colonne non spaziali nel file di output dBASE. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, vengono utilizzati i nomi derivati dall’istruzione SELECT.

Se viene indicato il parametro, ma i nomi delle colonne non vengono racchiusi tra due punti interrogativi, i nomi delle colonne verranno convertiti in caratteri maiuscoli. Il numero delle colonne specificate dovrà corrispondere al numero di colonne restituite dall’istruzione SELECT, in base alle indicazioni del parametro *istruzione_selezione*, fatta eccezione per la colonna spaziale.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

istruzione_selezione

Indica la selezione secondaria che restituisce i dati da esportare. La serie secondaria dovrà riportare esattamente una colonna spaziale e un numero qualsiasi di colonne di attributi. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

file_messaggi

Indica il nome completo del file (sulla macchina server) in cui verranno inseriti i messaggi relativi all’esportazione. Sebbene sia obbligatorio specificare un

ST_export_shape

valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il file dei messaggi per DB2 Spatial Extender non verrà creato.

I messaggi inviati a questo file, possono essere:

- Messaggi informativi, ad esempio il riepilogo dell'operazione di esportazione
- Messaggi di errore relativi ai dati che non è stato possibile esportare, ad esempio dati appartenenti a sistemi di coordinate diversi

La procedura memorizzata, che viene eseguita come un processo appartenente al proprietario dell'istanza DB2, deve disporre dei privilegi necessari per creare i file sul server.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Note di utilizzo:

E' possibile esportare solo una colonna spaziale alla volta.

La procedura memorizzata ST_export_shape crea o modifica i quattro file riportati di seguito:

- Il file shape principale (estensione .shp).
- Il file shape indice (estensione .shx).
- Un file dBASE che contiene i dati relativi alle colonne non spaziali (estensione .dbf). Questo file viene creato solo se è necessario esportare le colonne di attributi.
- Un file di proiezione che indica il sistema di coordinate associato ai dati spaziali, se il sistema non è uguale a "UNSPECIFIED" (estensione .prj). Il sistema di coordinate è ricavato dal primo record spaziale. Se record successivi hanno sistemi di coordinate diversi, si verifica un errore.

La seguente tabella descrive le modalità di memorizzazione dei dati DB2 nei file di attributi dBASE. Tutti gli altri tipi di dati DB2 non sono supportati.

Tabella 29. Memorizzazione dei tipi di dati DB2 nei file di attributi

Tipo SQL	Tipo .dbf	Lunghezza .dbf	Decimali .dbf	Commenti
SMALLINT	S	6	0	
INTEGER	S	11	0	
BIGINT	S	20	0	
DECIMAL	S	precisione+2	scale	
Da REAL FLOAT(1) a FLOAT(24)	F	14	6	
Da DOUBLE FLOAT(25) a FLOAT(53)	F	19	9	
CHARACTER, VARCHAR, LONG VARCHAR e DATALINK	C	len	0	lunghezza ≤ 255
DATE	D	8	0	
TIME	C	8	0	
TIMESTAMP	C	26	0	

Tutti i sinonimi utilizzati per i tipi di dati e per i tipi distinti, basati sui tipi elencati nella tabella precedente, sono supportati.

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_export_shape. Per esportare tutte le righe dalla tabella CUSTOMERS nel file shape /tmp/export_file, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_export_shape('/tmp/export_file',0,NULL,
    'select * from customers','/tmp/export_msg',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- “ST_import_shape” a pagina 257

ST_import_shape

Utilizzare questa procedura memorizzata per importare un file shape in un database abilitato per le operazioni spaziali. Utilizzando il parametro *indicatore_crea_tabella*, la procedura può funzionare in due modi:

- DB2 Spatial Extender crea una tabella con una colonna spaziale e colonne di attributi, quindi inserisce i dati del file nelle colonne della tabella.
- In alternativa, i dati shape e di attributo vengono caricati in una tabella esistente che dispone di una colonna spaziale e di colonne di attributi che corrispondono ai dati del file.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_import_shape.

memorizzata, che viene eseguita come un processo appartenente al proprietario dell'istanza DB2, deve disporre dei privilegi necessari per la lettura dei file sul server.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

colonne_attr_input

Indica un elenco di colonne di attributi da importare dal file dBASE. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, verranno importate tutte le colonne. Se il file dBASE non esiste, il parametro dovrà essere una stringa vuota o un valore nullo.

Per indicare un valore non nullo per il parametro, utilizzare una delle seguenti specifiche:

- **Elencare i nomi delle colonne di attributo.** L'esempio seguente mostra come indicare l'elenco dei nomi delle colonne di attributi da importare dal file dBASE:

```
N(COLONNA1,COLONNA5,COLONNA3,COLONNA7)
```

Se il nome della colonna non è racchiuso tra doppi apici, viene convertito in caratteri maiuscoli. Ciascun nome dell'elenco dovrà essere separato da una virgola. Tutti i nomi dovranno corrispondere ai nomi contenuti nella colonna nel file dBASE.

- **Elencare i numeri delle colonne di attributo.** L'esempio seguente mostra come indicare l'elenco dei numeri delle colonne di attributi da importare dal file dBASE:

```
P(1,5,3,7)
```

Le colonne sono numerate a partire da 1. Ciascun numero dell'elenco dovrà essere separato da una virgola.

- **Indicare di non importare i dati di attributo.** Indicare "", ovvero una stringa vuota che specifica esplicitamente a DB2 Spatial Extender di *non* importare i dati di attributo.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

nome_srs

Identifica il sistema di riferimento spaziale da utilizzare per le forme geometriche che verranno importate nella colonna spaziale. Il valore del parametro non può essere nullo.

La colonna spaziale non verrà registrata. E' necessario che il sistema di riferimento spaziale (SRS) esista già prima dell'importazione dei dati. Il processo di importazione non crea implicitamente il sistema di riferimento, ma confronta le coordinate del sistema di riferimento spaziale al sistema di coordinate specificato nel file .prj (se disponibile insieme al file shape). Il processo di importazione inoltre verifica che sia possibile rappresentare l'estensione dei dati nel sistema di riferimento spaziale fornito. In altre parole, il processo di importazione controlla che le estensioni rientrino tra i valori minimo e massimo delle coordinate X, Y, Z, ed M del sistema di riferimento spaziale.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

schema_tabella

Indica il nome dello schema a cui appartiene la tabella indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella in cui verrà caricato il file shape importato. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

tabelle_attr_input

Indica i nomi delle colonne della tabella in cui sono memorizzati i dati di attributo provenienti dal file dBASE. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, vengono utilizzati i nomi delle colonne presenti nel file dBASE.

Se il parametro viene specificato, il numero di nomi dovrà corrispondere al numero di colonne importate dal file dBASE. Se la tabella esiste già, le definizioni delle colonne dovranno corrispondere ai dati in entrata. La sezione "Note di utilizzo" a pagina 264 spiega come i dati di attributo vengono associati ai dati DB2.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

indicatore_crea_tabella

Indica se il processo di importazione dovrà creare una nuova tabella. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. La tabella viene creata se il parametro è nullo, oppure se viene utilizzato un qualsiasi valore diverso da zero. Se la tabella esiste già, viene restituito un messaggio d'errore. Se il parametro è 0 (zero), non viene creata alcuna tabella, quindi è necessario che la tabella sia già stata creata.

Il tipo di dati utilizzato con questo parametro è INTEGER.

parametri_creazione_tabella

Indica le opzioni da aggiungere all'istruzione CREATE TABLE per la creazione di una tabella in cui importare i dati. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, all'istruzione CREATE TABLE non verrà aggiunta alcuna opzione.

Per indicare le opzioni CREATE TABLE, utilizzare la sintassi dell'istruzione DB2 CREATE TABLE. Ad esempio, per indicare un tablespace in cui creare le tabelle, specificare:

```
IN tsName INDEX IN indexTsName LONG IN longTsName
```

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

colonna_spaziale

Il nome della colonna spaziale nella tabella in cui verranno inseriti i dati shape. Il valore del parametro non può essere nullo.

Nel caso di una nuova tabella, il parametro indica il nome della nuova colonna spaziale che verrà creata. In caso contrario, il parametro indica il nome di una colonna spaziale esistente nella tabella.

Il valore *colonna_spaziale* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

tipo_schema

Indica il nome dello schema del tipo di dati spaziali (indicato dal parametro *tipo_nome*) da utilizzare durante la creazione della colonna spaziale nella nuova tabella. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore DB2GSE.

Il valore *tipo_schema* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

tipo_nome

Indica il nome del tipo di dati da utilizzare per i valori spaziali. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il tipo di dati viene determinato dal file shape e sarà uno tra i seguenti:

- ST_Point
- ST_MultiPoint
- ST_MultiLineString
- ST_MultiPolygon

I file shape, per definizione, consentono di distinguere solo tra punti e punti multipli e non tra poligoni e multipoligoni o tra linee e linee multiple.

Se si desidera importare i dati in una tabella non ancora esistente, il tipo di dati verrà utilizzato anche per i dati della colonna spaziale. In tal caso, il tipo di dati potrà essere un super tipo di ST_Point, ST_MultiPoint, ST_MultiLineString o ST_MultiPolygon.

Il valore *tipo_nome* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

lunghezza_linea

Indica, il numero massimo di byte da assegnare alla colonna spaziale di una nuova tabella. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà utilizzata alcuna opzione esplicita INLINE LENGTH nell'istruzione CREATE TABLE e verranno utilizzati i valori predefiniti di DB2.

I record spaziali che superano questa dimensione vengono memorizzati separatamente nel tablespace LOB, i cui tempi di accesso potrebbero risultare più lunghi.

Di seguito sono riportate le dimensioni tipiche necessarie per i diversi tipi spaziali:

- **Un punto:** 292.
- **Punti multipli, riga o poligono:** Il valore più elevato possibile. Tenere presente che il numero totale di byte presenti in una riga non può superare il limite della dimensione di pagina del tablespace per il quale la tabella è stata creata.

Il valore viene descritto dettagliatamente nella documentazione di DB2 relativa all'istruzione CREATE TABLE SQL. Per determinare il numero di forme geometriche interne per le tabelle esistenti e modificare la lunghezza interna, fare riferimento all'utilità db2dart.

Il tipo di dati utilizzato con questo parametro è INTEGER.

id_colonna

Indica il nome della colonna da creare affinché contenga un numero univoco per ciascuna riga di dati. Gli strumenti ESRI richiedono una colonna chiamata SE_ROW_ID. I valori univoci per la colonna vengono generati automaticamente durante il processo di importazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo se nella tabella non esiste alcuna colonna (con un ID univoco in ciascuna riga), oppure se non si desidera aggiungere tale colonna alla nuova tabella. Se il parametro è nullo, i numeri univoci non verranno inseriti in nessuna colonna.

Limitazione: Non è possibile specificare un nome *id_colonna* che corrisponde al nome di una delle colonne del file dBASE.

I requisiti e gli effetti di questo parametro dipendono dall'esistenza della tabella.

- **Se la tabella esiste**, i dati del parametro *id_colonna* potranno essere valori interi di qualsiasi tipo (INTEGER, SMALLINT o BIGINT).
- **Se si desidera creare una nuova tabella**, la procedura memorizzata aggiungerà la colonna alla tabella durante il processo di creazione. La colonna viene definita come segue:

```
INTEGER NOT NULL PRIMARY KEY
```

Se il valore del parametro *id_colonna_identità* non è nullo ed è diverso da 0 (zero), la definizione viene ampliata come segue:

```
INTEGER NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY  
( START WITH 1 INCREMENT BY 1 )
```

Il valore *id_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

id_colonna_identità

Indica se l'*id_colonna* specificato deve essere creato utilizzando la clausola IDENTITY. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è 0 (zero) o nullo, la colonna non viene creata come colonna di identità. Se il parametro ha un qualsiasi valore diverso da 0 e non nullo, la colonna viene creata come colonna di identità. Il parametro viene ignorato dalle tabelle già esistenti.

Il tipo di dati utilizzato con questo parametro è SMALLINT.

riavvia_conteggio

Indica che l'operazione di importazione deve essere avviata al record $n + 1$. I primi n record vengono ignorati. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, verranno importati tutti i record (che cominciano con 1).

Il tipo di dati utilizzato con questo parametro è INTEGER.

ambito_commit

Indica di eseguire un COMMIT dopo l'importazione di n record. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, viene utilizzato il valore 0 (zero) e nessun record viene sottoposto a commit.

Il tipo di dati utilizzato con questo parametro è INTEGER.

file_eccezioni

Indica il nome completo del file shape in cui memorizzare i dati shape che non è stato possibile importare. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà creato alcun file.

Se si desidera specificare un valore per il parametro ed includere l'estensione del file, indicare .shp o .SHP. Se l'estensione è nulla, viene aggiunta l'estensione .shp.

Il file di eccezioni contiene l'intero blocco di righe per il quale l'istruzione di inserimento ha avuto esito negativo. Ad esempio, se l'importazione di una riga non riesce perché i dati shape hanno una codifica non corretta, e l'istruzione di inserimento tenta di importare 20 righe inclusa la riga contenente l'errore, l'intero blocco di 20 righe viene inserito nel file di eccezioni.

I record vengono inseriti nel file di eccezioni quando non è possibile indentificarli correttamente, come nel caso dei dati shape non validi. Certi tipi di errori nei file shape (file .shp) e indici shape (file .shx) non consentono una corretta identificazione dei record. In questo caso, nessun record viene inserito nel file di eccezioni, ma viene generato un messaggio che riporta il problema.

Specificando un valore per il parametro, sul server vengono creati quattro file. Per ulteriori informazioni sui file, consultare la sezione "Note di utilizzo" a pagina 264. La procedura memorizzata, che viene eseguita come un processo appartenente al proprietario dell'istanza DB2, deve disporre dei privilegi necessari per creare i file sul server. Se il file esiste già, la procedura restituisce un errore.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

file_messaggi

Indica il nome completo del file (sulla macchina server) in cui verranno inseriti i messaggi relativi all'importazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il file dei messaggi per DB2 Spatial Extender non verrà creato.

I messaggi inseriti in questo file, possono essere:

- Messaggi informativi, ad esempio il riepilogo dell'operazione di importazione
- Messaggi di errore relativi ai dati che non è stato possibile importare, ad esempio dati appartenenti a sistemi di coordinate diversi

I messaggi corrispondono ai dati shape memorizzati nel file di eccezioni (identificato dal parametro *file_eccezioni*).

ST_import_shape

La procedura memorizzata, che viene eseguita come un processo appartenente al proprietario dell'istanza DB2, deve disporre dei privilegi necessari per creare i file sul server. Se il file esiste già, la procedura restituisce un errore.

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Note di utilizzo:

La procedura memorizzata ST_import_shape utilizza da uno a quattro file:

- Il file shape principale (estensione .shp). Il file è obbligatorio.
- Il file shape indice (estensione .shx). Il file è facoltativo. Se presente, le prestazioni dell'operazione di importazione potrebbero essere migliori.
- Un file dBASE che contiene i dati di attributo (estensione .dbf). Il file è obbligatorio solo se si desidera importare dati di attributo.
- Il file di proiezione che indica il sistema di coordinate dei dati shape (estensione .prj). Il file è facoltativo. Se presente, il sistema di coordinate definito, viene confrontato con il sistema di coordinate del sistema di riferimento spaziale indicato dal parametro *srs_id*.

La seguente tabella mostra come i tipi di dati di attributo dBASE vengono associati ai tipi di dati DB2. Tutti gli altri tipi di dati di attributo non sono supportati.

Tabella 30. Relazione tra tipi di dati DB2 e tipi di dati di attributo dBASE

Tipo .dbf	Lunghezza .dbfb (Vedere nota)	Decimali .dbfb (Vedere nota)	Tipo SQL	Commenti
S	< 5	0	SMALLINT	
S	< 10	0	INTEGER	
S	< 20	0	BIGINT	
S	<i>len</i>	<i>dec</i>	DECIMAL(<i>len,dec</i>)	<i>len</i> <32
F	<i>len</i>	<i>dec</i>	REAL	<i>len + dec</i> < 7
F	<i>len</i>	<i>dec</i>	DOUBLE	
C	<i>len</i>		CHAR(<i>len</i>)	
L			CHAR(1)	

Tabella 30. Relazione tra tipi di dati DB2 e tipi di dati di attributo dBASE (Continua)

Tipo .dbf	Lunghezza .dbfb (Vedere nota)	Decimali .dbfb (Vedere nota)	Tipo SQL	Commenti
D			DATE	

Nota: La tabella contiene le seguenti variabili, entrambe definite nell'intestazione del file dBASE:

- *len*, che rappresenta la lunghezza della colonna nel file dBASE. DB2 Spatial Extender utilizza questo valore per due scopi:
 - Definire il livello di precisione per il tipo di dati SQL, DECIMAL, oppure la lunghezza per il tipo di dati SQL, CHAR
 - Stabilire se utilizzare il tipo intero o a virgola mobile
- *dec*, che rappresenta il numero massimo di cifre a destra della virgola decimale per la colonna nel file dBASE. DB2 Spatial Extender utilizza questo valore per definire la scala per il tipo di dati SQL, DECIMAL.

Ad esempio, se il file dBASE contiene una colonna di dati la cui lunghezza (*len*) è definita come 20 e il numero di cifre a destra della virgola decimale (*dec*) è definita come 5, quando DB2 Spatial Extender importa i dati da quella determinata colonna, utilizzerà i valori di *len* e *dec* per ottenere il tipo di dati SQL, DECIMAL(20,5).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_import_shape. Per importare il file shape /tmp/officesShape nella tabella OFFICES, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_import_shape('/tmp/officesShape',NULL,'USA_SRS_1',NULL,
                             'OFFICES',NULL,0,NULL,'LOCATION',NULL,NULL,NULL,
                             NULL,NULL,NULL,NULL,'/tmp/import_msg',?,?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_export_shape" a pagina 254

ST_register_geocoder

Utilizzare questa procedura memorizzata per registrare un geocoder diverso da DB2SE_USA_GEOCODER, fornito con DB2 Spatial Extender. Il geocoder DB2SE_USA_GEOCODER viene registrato da DB2 Spatial Extender nel momento in cui il database viene abilitato.

Prerequisiti: Prima di registrare un geocoder, procedere come segue:

- Assicurarsi che le funzioni che implementano il geocoder siano già state create. E' possibile registrare ciascuna funzione di geocoding come un geocoder con nome univoco.
- Richiedere le seguenti informazioni al fornitore del geocoder:
 - L'istruzione SQL che crea la funzione

ST_register_geocoder

- I valori da utilizzare con i parametri ST_create_srs affinché i dati geometrici vengano supportati
- Le informazioni per registrare il geocoder, quali:
 - La descrizione del geocoder
 - Le descrizioni dei parametri del geocoder
 - I valori predefiniti dei parametri del geocoder

Il tipo di risultati della funzione di geocoding deve corrispondere al tipo di dati della colonna sottoposta a geocoding. I parametri di geocoding possono essere costituiti dal nome di una colonna (definita *colonna di geocoding*) che contiene i dati necessari al geocoder. Ad esempio, i parametri del geocoder possono identificare indirizzi oppure valori significativi per il geocoder, ad esempio il valore minimo di corrispondenza. Se il parametro di geocoding è il nome di una colonna, la colonna dovrà essere situata nella stessa tabella o vista a cui appartiene la colonna sottoposta a geocoding.

Il tipo di risultati della funzione di geocoding viene utilizzato come tipo di dati della colonna sottoposta a geocoding. Il tipo di risultato può essere un tipo di dati DB2, un tipo di dati definito dall'utente o un tipo strutturato. Se viene restituito un tipo di dati definito dall'utente o strutturato, la funzione di geocoding dovrà restituire un valore valido per ciascun tipo di dati. Se la funzione di geocoding restituisce valori di tipo spaziale, ad esempio ST_Geometry o uno dei relativi tipi secondari, la funzione di geocoding creerà una forma geometrica valida. La forma geometrica viene rappresentata mediante un sistema di riferimento spaziale esistente. La forma geometrica viene considerata valida se, richiamando la funzione spaziale ST_IsValid, viene restituito il valore 1. I dati restituiti dalla funzione di geocoding vengono aggiornati o inseriti nella colonna sottoposta a geocoding, in base all'operazione (INSERT o UPDATE) utilizzata per generare i valori.

Per determinare se un geocoder è già registrato, fare riferimento alla vista del catalogo DB2GSE.ST_GEOCODERS.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_register_gc.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database che contiene il geocoder registrato utilizzando questa procedura.

Sintassi:

```
►► db2gse.ST_register_geocoder ( ( nome_geocoder , schema_funzione , schema_funzione )
                                ( nome_funzione , nome_specifico , nome_funzione )
                                ( valori_parametro_predefiniti , descrizione_parametri , nome_funzione )
                                ( fornitore , descrizione ) )
```

Descrizioni dei parametri:

nome_geocoder

Identifica il geocoder. Il valore del parametro non può essere nullo.

Il valore *nome_geocoder* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

schema_funzione

Indica il nome dello schema per la funzione che implementa il geocoder.

Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la funzione viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_funzione* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_funzione

Indica il nome non qualificato della funzione che implementa il geocoder. E' necessario che la funzione sia già stata creata ed elencata in SYSCAT.ROUTINES.

Se il parametro *nome_specifico* è stato specificato, questo parametro potrà avere valore nullo. Se il parametro *nome_specifico* non è stato specificato, il valore *nome_funzione*, insieme al valore *schema_funzione* definito implicitamente o esplicitamente, dovranno identificare unicamente la funzione. Se il parametro *nome_funzione* non viene specificato, DB2 Spatial Extender ricaverà il valore di *nome_funzione* dalla vista del catalogo SYSCAT.ROUTINES.

Il valore *nome_funzione* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_specifico

Indica il nome specifico della funzione che implementa il geocoder. E' necessario che la funzione sia già stata creata ed elencata in SYSCAT.ROUTINES.

Se il parametro *nome_funzione* è stato specificato e la combinazione di *schema_funzione* e *nome_funzione* identifica unicamente la funzione di geocoding, è possibile indicare un valore nullo per questo parametro. Se il nome della funzione di geocoding è sovraccarico, il parametro *nome_specifico* non potrà avere valore nullo. Il nome di una funzione risulta *sovraccarico* se corrisponde al nome di una o più funzioni aventi parametri o tipi di dati diversi.

Il valore *nome_specifico* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

valori_parametro_predefiniti

Indica l'elenco dei valori dei parametri di geocoding predefiniti per la funzione di geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se tutti i valori del parametro *valori_parametro_predefiniti* sono nulli, i valori predefiniti di tutti i parametri saranno nulli.

ST_register_geocoder

Se si desidera indicare i valori dei parametri, specificarli nell'ordine in cui sono definiti dalla funzione e separare i nomi con una virgola. Ad esempio:

```
valore_param1_predef, valore_param2_predef, ...
```

Ciascun valore è una espressione SQL. Attenersi a queste istruzioni:

- Se il valore è una stringa, racchiudere la stringa tra apici.
- Se il valore è un numero, non racchiudere il valore tra apici.
- Se il valore del parametro è nullo, associarlo al tipo corretto. Ad esempio, invece di specificare NULL, indicare
CAST(NULL AS INTEGER)
- Se il parametro di geocoding deve corrispondere a una colonna di geocoding, non indicare il valore del parametro predefinito.

Se il valore di un parametro non viene specificato, ovvero vengono utilizzate due virgole consecutive (...), il parametro dovrà essere specificato durante l'impostazione della funzione di geocoding oppure durante l'esecuzione della funzione in modalità batch con il parametro *valori_parametro* delle rispettive procedure memorizzate.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

descrizione_parametri

Indica l'elenco dei valori dei parametri di geocoding per la funzione di geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo.

Se tutti i valori del parametro *descrizione_parametri* sono nulli, tutte le descrizioni dei parametri saranno nulle. La descrizione di ciascun parametro indica lo scopo e l'utilizzo del parametro e può essere costituita da un massimo di 256 caratteri. Le descrizioni dei parametri devono essere separate da virgole e dovranno essere nell'ordine in cui i parametri sono definiti dalla funzione. Se nella descrizione si desidera utilizzare una virgola come segno di punteggiatura, racchiudere la stringa tra singoli o doppi apici. Ad esempio:
descrizione, 'descrizione2, che contiene una virgola', descrizione3

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

fornitore

Indica il nome del fornitore del geocoder. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà registrata alcuna informazione sul fornitore che ha implementato il geocoder.

Il tipo di dati utilizzato con questo parametro è VARCHAR(128).

descrizione

Descrive il geocoder descrivendo la relativa applicazione. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, non verrà registrata alcuna informazione sul geocoder.

Suggerimento: Inserire le seguenti informazioni nella descrizione:

- Il nome del sistema di coordinate se si desidera che il risultato siano dati spaziali, ad esempio WKT (well-known text) o WKB (well-known binary)
- Il sistema di riferimento spaziale, se si desidera che il risultato sia ST_Geometry o uno dei tipi secondari
- Il nome dell'area geografica a cui si riferisce il geocoder

- Tutte le altre informazioni sul geocoder necessarie

Il tipo di dati utilizzato con questo parametro è VARCHAR(256).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio mostra come creare un geocoder che esegua il geocoding dei valori di latitudine e longitudine per convertirli in dati spaziali ST_Point. Come prima operazione, creare la funzione *lat_long_gc_func*, quindi registrare il geocoder SAMPLEGC, che utilizzerà la funzione *lat_long_gc_func*.

Di seguito viene fornito un esempio di istruzione SQL che è possibile utilizzare per creare la funzione *lat_long_gc_func* che restituisce ST_Point:

```
CREATE FUNCTION lat_long_gc_func(latitude double,
    longitude double, srId integer)
    RETURNS db2gse.ST_Point
    LANGUAGE SQL
    RETURN db2gse.ST_Point(latitude, longitude, srId)
```

Dopo aver creato la funzione, è possibile registrarla come geocoder. Nell'esempio seguente, per richiamare la procedura memorizzata *ST_register_geocoder* affinché registri il geocoder SAMPLEGC, con la funzione *lat_long_gc_func*, viene utilizzato il processore riga comandi DB2 CALL:

```
call db2gse.ST_register_geocoder ('SAMPLEGC',NULL,'LAT_LONG_GC_FUNC','',1'
    ,NULL,'My Company','Latitude/Longitude to
    ST_Point Geocoder'?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- “ST_unregister_geocoder” a pagina 280

ST_register_spatial_column

Utilizzare questa procedura memorizzata per registrare una colonna spaziale ed associarvi un sistema spaziale di riferimento (SRS). Durante l'elaborazione della procedura, le informazioni sulla colonna spaziale da registrare vengono aggiunte alla vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS. Affinché tutte le forme geometriche utilizzino lo stesso SRS, durante la registrazione della colonna spaziale, nella tabella viene creato un vincolo.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_register_layer.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella a cui appartiene la colonna spaziale da registrare
- Privilegio CONTROL o ALTER per la tabella

Sintassi:

```

▶▶ db2gse.ST_register_spatial_column—(—schema_tabella—, —nome_tabella—▶
└──┬───┘
  nullable
▶, —nome_colonna—, —nome_srs—)▶▶

```

Descrizioni dei parametri:

schema_tabella

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella o vista che contiene la colonna da registrare. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica la colonna da registrare. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_srs

Nome del sistema di riferimento spaziale da utilizzare per la colonna spaziale. Il valore del parametro non può essere nullo.

Il valore *nome_srs* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:*codice_msg*

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_register_spatial. Per registrare la colonna spaziale LOCATION nella tabella CUSTOMERS, viene utilizzato il comando DB2 CALL. Il comando CALL indica USA_SRS_1 come valore del parametro *nome_srs*:

```
call db2gse.ST_register_spatial_column(NULL,'CUSTOMERS','LOCATION',
    'USA_SRS_1',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- “ST_unregister_spatial_column” a pagina 281

ST_remove_geocoding_setup

Utilizzare questa procedura memorizzata per eliminare tutte le impostazioni della colonna sottoposta a geocoding.

La procedura elimina dalle viste dei cataloghi DB2GSE.ST_GEOCODING e DB2GSE.ST_GEOCODING_PARAMETERS le informazioni associate alla colonna specificata.

Limitazione: Non è possibile eliminare le impostazioni di geocoding se per la colonna è stata abilitata la funzione di geocoding automatica.

ST_remove_geocoding_setup

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verrà eseguito il geocoder specificato
- Privilegio CONTROL o UPDATE per questa tabella

Sintassi:

```
►► db2gse.ST_remove_geocoding_setup—(—schema_tabella—, —nome_tabella—►  
                                  └─┬─┘  
                                  nullo  
►, —nome_colonna—) —►►
```

Descrizioni dei parametri:

schema_tabella

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella o vista che contiene la colonna in cui il geocoder verrà inserito o aggiornato. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica il nome della colonna in cui verranno inseriti o aggiornati i dati sottoposti a geocoding. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) oppure, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_remove_geocoding. Per eliminare le impostazioni di geocoding dalla colonna LOCATION della tabella CUSTOMER, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_remove_geocoding_setup(NULL, 'CUSTOMERS', 'LOCATION',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_setup_geocoding" a pagina 276

ST_run_geocoding

Utilizzare questa procedura memorizzata per eseguire un geocoder in modalità batch in una colonna sottoposta a geocoding.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_run_gc.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verrà eseguito il geocoder specificato
- Privilegio CONTROL o UPDATE per questa tabella

Sintassi:

```

▶▶ db2gse.ST_run_geocoding—(—schema_tabella—, —nome_tabella—, —————▶
                             |nonnullo|
▶ —nome_colonna—, —nome_geocoder—, —valori_parametro—, —————▶
                             |nonnullo| |nonnullo|
▶ —clausola_where—, —ambito_commit—) —————▶▶
                             |nonnullo| |nonnullo|

```

Descrizioni dei parametri:

schema_tabella

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo

ST_run_geocoding

parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella o vista che contiene la colonna in cui il geocoder verrà inserito o aggiornato. Se si desidera specificare il nome di una vista, è necessario che tale vista sia aggiornabile. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica il nome della colonna in cui verranno inseriti o aggiornati i dati sottoposti a geocoding. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_geocoder

Indica il nome del geocoder che eseguirà il geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il geocoding verrà eseguito dal geocoder specificato durante l'impostazione.

Il valore *nome_geocoder* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

valori_parametro

Indica l'elenco dei valori dei parametri di geocoding per la funzione di geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se tutti i valori del parametro *valori_parametro* sono nulli, verranno utilizzati i valori indicati durante l'impostazione del geocoder, oppure i valori predefiniti se il geocoder non è stato impostato.

Se si desidera indicare i valori dei parametri, specificarli nell'ordine in cui sono definiti dalla funzione e separare i nomi con una virgola. Ad esempio:

parametro1-valore,parametro2-valore,...

I valori del parametro possono essere nomi di colonna, stringhe, valori numerici o nulli.

Ciascun valore è una espressione SQL. Attenersi a queste istruzioni:

- Se il valore del parametro è il nome di una colonna di geocoding, verificare che la colonna appartenga alla stessa tabella o vista in cui si trova la colonna sottoposta a geocoding.

- Se il valore del parametro è una stringa, racchiudere la stringa tra apici.
- Se il valore è un numero, non racchiudere il valore tra apici.
- Se il valore del parametro è nullo, associarlo al tipo corretto. Ad esempio, invece di specificare NULL, indicare
CAST(NULL AS INTEGER)

Se il valore di un parametro non viene specificato, ovvero vengono utilizzate due virgole consecutive (...), il parametro dovrà essere specificato durante l'impostazione della funzione di geocoding oppure durante l'esecuzione della funzione in modalità batch con il parametro *valori_parametro* delle rispettive procedure memorizzate.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

clausola_where

Indica il testo della clausola the WHERE, che definisce una limitazione nella serie di record da sottoporre a geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo.

Se il parametro *clausola_where* è nullo, il risultato è diverso se l'operazione di geocoding è stata impostata o meno per la colonna (specificata con il parametro *nome_colonna*) prima dell'esecuzione della procedura memorizzata. Se il parametro *clausola_where* è nullo:

- Per definire il parametro *clausola_where* verrà utilizzato il valore specificato durante l'impostazione della funzione di geocoding.
- Se la funzione di geocoding non è stata impostata o non è stato specificato alcun valore durante l'impostazione, non verrà utilizzata alcuna clausola.

E' possibile specificare una clausola che fa riferimento a qualsiasi colonna della tabella o della vista in cui verrà eseguito il geocoder. Non specificare la parola chiave WHERE.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

ambito_commit

Indica di eseguire un COMMIT dopo l'esecuzione del geocoding per *n* record. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo.

Se il parametro *ambito_commit* è nullo, il risultato è diverso se l'operazione di geocoding è stata impostata o meno per la colonna (specificata con il parametro *nome_colonna*) prima dell'esecuzione della procedura memorizzata. Se il parametro *ambito_commit* è nullo:

- Per definire il parametro *ambito_commit* verrà utilizzato il valore specificato per la colonna durante l'impostazione della funzione di geocoding.
- Se la funzione di geocoding non è stata impostata o non è stato specificato alcun valore durante l'impostazione, viene utilizzato il valore predefinito 0 (zero) e non verrà eseguito alcun COMMIT.

Il tipo di dati utilizzato con questo parametro è INTEGER.

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto

ST_run_geocoding

esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_run_geocoding. Per eseguire la funzione di geocoding della colonna LOCATION della tabella CUSTOMER, viene utilizzato il comando DB2 CALL: Il comando CALL specifica DB2SE_USA_GEOCODER come valore del parametro *nome_geocoder* e 10 come valore del parametro *ambito_commit*. Il COMMIT verrà eseguito dopo il geocoding di ogni 10 record:

```
call db2gse.ST_run_geocoding(NULL, 'CUSTOMERS', 'LOCATION',  
                             'DB2SE_USA_GEOCODER',NULL,NULL,10,?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_setup_geocoding" a pagina 276

ST_setup_geocoding

Utilizzare questa procedura memorizzata per associare una colonna da sottoporre a geocoding a un geocoder ed impostare i valori dei parametri di geocoding corrispondenti. Le impostazioni vengono registrate nelle viste dei cataloghi DB2GSE.ST_GEOCODING e DB2GSE.ST_GEOCODING_PARAMETERS.

La procedura non richiama la funzione di geocoding, ma consente di specificare le impostazioni dei parametri per la colonna da sottoporre a geocoding. Effettuando queste impostazioni, sarà possibile richiamare le funzioni di geocoding in modalità batch o automatica attraverso interfacce più semplici. Le impostazioni dei parametri sostituiranno tutti i valori predefiniti dei parametri relativi al geocoder, specificati durante la registrazione del geocoder. E' possibile sovrascrivere tali impostazioni anche eseguendo la procedura memorizzata ST_run_geocoding in modalità batch.

Questa operazione è uno dei prerequisiti per la funzione di geocoding automatica. Non sarà possibile eseguire la funzione di geocoding automatica senza aver impostato i parametri di geocoding. Questa operazione non è richiesta per la funzione di geocoding in modalità batch che può essere eseguita con o senza aver eseguito le fasi di impostazione. Tuttavia, se le impostazioni vengono effettuate prima di eseguire la funzione di geocoding in modalità batch, e valori dei parametri non vengono specificati al momento dell'esecuzione, verranno utilizzati quelli indicati durante l'impostazione.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verrà eseguito il geocoder specificato
- Privilegio CONTROL o UPDATE per questa tabella

Sintassi:

```

▶▶ db2gse.ST_setup_geocoding—(—schema_tabella—, —nome_tabella—, —
      |nessuno—)
▶ —nome_colonna—, —nome_geocoder—, —valori_parametro—, —
      |nessuno—)
▶ (colonne_autogeocoding—, —clausola_where—, —ambito_commit—)
      |nessuno—, —nessuno—, —nessuno—)

```

Descrizioni dei parametri:*schema_tabella*

Il nome dello schema a cui appartiene la tabella o vista indicata nel parametro *nome_tabella*. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, come nome dello schema per la tabella o vista viene utilizzato il valore del registro speciale CURRENT SCHEMA.

Il valore *schema_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_tabella

Indica il nome non qualificato della tabella o vista che contiene la colonna in cui il geocoder verrà inserito o aggiornato. Se si desidera specificare il nome di una vista, è necessario che tale vista sia aggiornabile. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica il nome della colonna in cui verranno inseriti o aggiornati i dati sottoposti a geocoding. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_geocoder

Indica il nome del geocoder che eseguirà il geocoding. Il valore del parametro non può essere nullo.

ST_setup_geocoding

Il valore *nome_geocoder* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

valori_parametro

Indica l'elenco dei valori dei parametri di geocoding per la funzione di geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se tutti i valori del parametro *valori_parametro* sono nulli, verranno utilizzati i valori predefiniti dei parametri indicati durante la registrazione del geocoder.

Se si desidera indicare i valori dei parametri, specificarli nell'ordine in cui sono definiti dalla funzione e separare i nomi con una virgola. Ad esempio:

parametro1-valore,parametro2-valore,...

Ciascun valore è una espressione SQL e può essere un nome di colonna, una stringa, un valore numerico o nullo. Attenersi a queste istruzioni:

- Se il valore del parametro è il nome di una colonna di geocoding, verificare che la colonna appartenga alla stessa tabella o vista in cui si trova la colonna sottoposta a geocoding.
- Se il valore del parametro è una stringa, racchiudere la stringa tra apici.
- Se il valore è un numero, non racchiudere il valore tra apici.
- Se il valore del parametro è nullo, associarlo al tipo corretto. Ad esempio, invece di specificare NULL, indicare
CAST(NULL AS INTEGER)

Se il valore di un parametro non viene specificato, ovvero vengono utilizzate due virgole consecutive (...), il parametro dovrà essere specificato durante l'impostazione della funzione di geocoding oppure durante l'esecuzione della funzione in modalità batch con il parametro *valori_parametro* delle rispettive procedure memorizzate.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

colonne_autogeocoding

Indica l'elenco di nomi di colonna in cui verrà creato il trigger. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo e la funzione di geocoding automatica è abilitata, il trigger viene attivato aggiornando una qualsiasi colonna nella tabella.

Se si desidera specificare un valore per il parametro *colonne_autogeocoding*, indicare i nomi delle colonne in qualsiasi ordine e separarli con una virgola. E' necessario che il nome della colonna sia presente nella stessa tabella che contiene la colonna sottoposta a geocoding.

Le impostazioni del parametro vengono applicate solo a funzioni di geocoding automatiche successive.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

clausola_where

Indica il testo della clausola the WHERE, che definisce una limitazione nella serie di record da sottoporre a geocoding. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, nella clausola WHERE non verrà definita alcuna limitazione.

La clausola può fare riferimento ad una qualsiasi colonna o vista della tabella in cui viene eseguito il geocoder. Non specificare la parola chiave WHERE.

Le impostazioni del parametro vengono applicate solo a funzioni di geocoding in modalità batch successive.

Il tipo di dati utilizzato con questo parametro è VARCHAR(32K).

ambito_commit

Indica di eseguire un COMMIT dopo l'esecuzione del geocoding per *n* record. Sebbene sia obbligatorio specificare un valore per questo parametro, tale valore può essere nullo. Se il parametro è nullo, il COMMIT verrà eseguito dopo che tutti i record sono stati sottoposti a geocoding.

Le impostazioni del parametro vengono applicate solo a funzioni di geocoding in modalità batch successive.

Il tipo di dati utilizzato con questo parametro è INTEGER.

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_setup_geocoding. Per impostare il processo di geocoding per la colonna LOCATION nella tabella CUSTOMER, viene utilizzato il comando DB2 CALL. Il comando CALL indica DB2SE_USA_GEOCODER come valore del parametro *nome_geocoder*:

```
call db2gse.ST_setup_geocoding(NULL, 'CUSTOMERS', 'LOCATION',
'DB2SE_USA_GEOCODER', 'ADDRESS,CITY,STATE,ZIP',1,100,80,,,'$HOME/sql1lib/
gse/refdata/ky.edg','$HOME/sql1lib/samples/spatial/EDGESample.loc',
'ADDRESS,CITY,STATE,ZIP',NULL,10,?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- “ST_unregister_geocoder” a pagina 280
- “ST_remove_geocoding_setup” a pagina 271

ST_unregister_geocoder

Utilizzare questa procedura memorizzata per annullare la registrazione di un geocoder diverso da DB2SE_USA_GEOCODER, fornito con DB2 Spatial Extender.

Limitazione: Non è possibile annullare la registrazione di un geocoder non specificato nell'impostazione del geocoding di nessuna colonna.

Per stabilire se un geocoder è specificato nell'impostazione della funzione di geocoding di una colonna, controllare le viste di catalogo DB2GSE.ST_GEOCODING e DB2GSE.ST_GEOCODING_PARAMETERS. Le informazioni sul geocoder di cui si desidera annullare la registrazione, sono contenute nella vista del catalogo DB2GSE.ST_GEOCODERS.

Questa procedura memorizzata sostituisce la procedura db2gse.gse_unregister_gc.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database che contiene il geocoder di cui si desidera annullare la registrazione.

Sintassi:

►►—db2gse.ST_unregister_geocoder—(—*nome_geocoder*—)—————►►

Descrizioni dei parametri:

nome_geocoder

Identifica il geocoder. Il valore del parametro non può essere nullo.

Il valore *nome_geocoder* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

ST_unregister_spatial_column

nome_tabella

Indica il nome non qualificato della tabella o vista che contiene la colonna indicata dal parametro *nome_colonna*. Il valore del parametro non può essere nullo.

Il valore *nome_tabella* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

nome_colonna

Indica la colonna di cui si desidera annullare la registrazione. Il valore del parametro non può essere nullo.

Il valore *nome_colonna* viene convertito in caratteri maiuscoli a meno che non sia racchiuso tra doppi apici.

Il tipo di dati da utilizzare con il parametro è VARCHAR(128) o, se il valore viene racchiuso tra doppi apici, VARCHAR(130).

Parametri di output:

codice_msg

Indica il codice del messaggio restituito dalla procedura memorizzata. Il valore del parametro di output indica se durante l'elaborazione della procedura si è verificata una condizione di errore, di avvertenza o se l'operazione ha avuto esito positivo. Se il valore del parametro indica esito positivo o avvertenza, significa che la procedura ha completato le attività. Se il valore del parametro indica una condizione di errore, non verrà applicata alcuna modifica al database.

Il tipo di dati utilizzato con questo parametro di output è INTEGER.

testo_msg

Indica il testo del messaggio corrispondente al codice del messaggio restituito dalla procedura memorizzata. Il testo del messaggio può includere ulteriori informazioni sull'esito positivo, l'avvertenza o su un'eventuale condizione di errore.

Il tipo di dati utilizzato con questo parametro di output è VARCHAR(1024).

Esempio:

Questo esempio indica come utilizzare il processore riga comandi DB2 per richiamare la procedura memorizzata ST_unregister_spatial_column. Per annullare la registrazione della colonna spaziale LOCATION nella tabella CUSTOMERS, viene utilizzato il comando DB2 CALL:

```
call db2gse.ST_unregister_spatial_column(NULL,'CUSTOMERS','LOCATION',?,?)
```

I due punti interrogativi nella parte finale del comando CALL rappresentano i parametri di output, *codice_msg* e *testo_msg*. I valori dei parametri vengono visualizzati dopo l'esecuzione della procedura memorizzata.

Riferimenti correlati:

- "ST_register_spatial_column" a pagina 270

Capitolo 21. Viste del catalogo

Le viste del catalogo di Spatial Extender contengono informazioni su:

“Vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS”

I sistemi di coordinati che è possibile utilizzare

“Vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS” a pagina 284

Le colonne spaziali in cui è possibile inserire o aggiornare i dati.

“Vista del catalogo DB2GSE.ST_GEOCODERS” a pagina 287 e “Vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS” a pagina 288

I geocoder che è possibile utilizzare

“Vista del catalogo DB2GSE.ST_GEOCODING” a pagina 287 e “Vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS” a pagina 288

Le specifiche per l'impostazione dell'esecuzione automatica del geocoder e per le impostazioni delle operazioni da eseguire durante l'esecuzione della funzione di geocoding in modalità batch.

“Vista del catalogo DB2GSE.ST_SIZINGS” a pagina 290

La lunghezza massima consentita dei valori che è possibile assegnare alle variabili.

“Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS” a pagina 291

I sistemi di riferimento spaziali che è possibile utilizzare.

“Vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE” a pagina 293

Le unità di misura (metri, miglia, piedi e così via) in cui è possibile esprimere le distanze generate dalle funzioni spaziali.

Vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS

Per ottenere informazioni sui sistemi di coordinate registrati, interrogare la vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS. Spatial Extender registra automaticamente i sistemi di coordinate nel catalogo di Spatial Extender quando:

- Si abilita un database per le operazioni spaziali.
- Gli utenti definiscono altri sistemi di coordinate al database.

La seguente tabella contiene una descrizione delle colonne contenute nella vista.

Tabella 31. Colonne nella vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS

Nome	Tipo di dati	Annullabile	Contenuto
COORDSYS_NAME	VARCHAR(128)	No	Nome del sistema di coordinate. Il nome è univoco nel database.

vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS

Tabella 31. Colonne nella vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
COORDSYS_TYPE	VARCHAR(128)	No	Il tipo associato a questo sistema di coordinate: PROIETTATO Bidimensionale. GEOGRAFICO Tridimensionale. Utilizza coordinate X e Y. GEOCENTRICO Tridimensionale. Utilizza coordinate X, Y e Z. NON SPECIFICATO Sistema di coordinate astratto o non reale. I valori di questa colonna vengono richiamati dalla colonna DEFINITION.
DEFINITION	VARCHAR(2048)	No	Rappresentazione di testo noto della definizione del sistema di coordinate.
ORGANIZATION	VARCHAR(128)	Sì	Nome dell'organizzazione (ad esempio, un ente standard come l'European Petrol Survey Group o l'ESPG) che ha definito il sistema di coordinate. Se la colonna ORGANIZATION_COORDSYS_ID non contiene alcun valore, questa colonna rimarrà vuota.
ORGANIZATION_COORDSYS_ID	INTEGER	Sì	Identificativo numerico assegnato a questo sistema di coordinate dall'organizzazione che ha definito il sistema. L'identificativo e il valore nella colonna ORGANIZATION identificano unicamente il sistema di coordinate, a meno che sia l'identificativo che il valore siano nulli. Se la colonna ORGANIZATION non contiene alcun valore, la colonna ORGANIZATION_COORDSYS_ID rimarrà vuota.
DESCRIPTION	VARCHAR(256)	Sì	Descrizione del sistema di coordinate che indica la propria applicazione.

Vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS

Utilizzare la vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS per ottenere informazioni su tutte le colonne spaziali nelle tabelle che contengono dati spaziali del database. Se una colonna spaziale è stata registrata come associata a un sistema di riferimento spaziale, è possibile utilizzare la vista per individuare il nome e l'identificativo numerico del sistema. Per ulteriori informazioni sulle colonne spaziali, eseguire un'interrogazione della vista del catalogo SYSCAT.COLUMN.

La seguente tabella contiene la descrizione della vista DB2GSE.ST_GEOMETRY_COLUMNS.

Tabella 32. Colonne nella vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS

Nome	Tipo di dati	Annullabile	Contenuto
TABLE_SCHEMA	VARCHAR(128)	No	Nome dello schema a cui appartiene la tabella che contiene questa colonna spaziale.
TABLE_NAME	VARCHAR(128)	No	Nome generico della tabella che contiene questa colonna spaziale.

Vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS

Tabella 32. Colonne nella vista del catalogo DB2GSE.ST_GEOMETRY_COLUMNS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
COLUMN_NAME	VARCHAR(128)	No	Nome della colonna spaziale. La combinazione di TABLE_SCHEMA, TABLE_NAME e COLUMN_NAME identificano unicamente questa colonna.
TYPE_SCHEMA	VARCHAR(128)	No	Nome dello schema a cui appartiene il tipo di dati dichiarato in questa colonna spaziale. Il nome viene richiamato dal catalogo del DB2.
TYPE_NAME	VARCHAR(128)	No	Nome generico del tipo di dati dichiarato di questa colonna spaziale. Il nome viene richiamato dal catalogo del DB2.
SRS_NAME	VARCHAR(128)	Sì	Nome del sistema di riferimento spaziale associato alla colonna spaziale. Se alla colonna non è stato associato alcun sistema di riferimento spaziale, il valore SRS_NAME sarà nullo.
SRS_ID	INTEGER	Sì	Identificativo numerico del sistema di riferimento spaziale associato a questa colonna spaziale. Se alla colonna non è stato associato alcun sistema di riferimento spaziale, il valore SRS_ID sarà nullo.

Vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS

Una volta abilitato un database per le operazioni spaziali, le informazioni sui parametri relativi al geocoder fornito, DB2GSE_USA_GEOCODER, vengono registrate nel catalogo DB2 Spatial Extender. Nel caso venissero registrati nuovi geocoder, le informazioni relative ai nuovi parametri verranno registrate nel catalogo. Per richiamare le informazioni relative ai parametri dei geocoder dal catalogo, interrogare la vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS. La seguente tabella contiene una descrizione delle colonne contenute nella vista.

Per ulteriori informazioni sui parametri dei geocoder, interrogare la vista del catalogo DB2, SYSCAT.ROUTINEPARMS. Per una descrizione di questa vista, fare riferimento a *Riferimenti SQL*.

Tabella 33. Colonne nella vista DB2GSE.ST_GEOCODER_PARAMETERS

Nome	Tipo di dati	Annullabile	Contenuto
GEOCODER_NAME	VARCHAR(128)	No	Nome del geocoder a cui appartengono questi parametri.

vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS

Tabella 33. Colonne nella vista DB2GSE.ST_GEOCODER_PARAMETERS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
ORDINAL	SMALLINT	No	<p>Posizione del parametro (ovvero il parametro indicato nella colonna PARAMETER_NAME) nella firma della funzione utilizzata come geocoder nella colonna GEOCODER_NAME.</p> <p>I valori combinati nelle colonne GEOCODER_NAME e ORDINAL identificano unicamente questo parametro.</p> <p>Le informazioni relative a questo parametro sono disponibili anche in un record nella vista del catalogo SYSCAT.ROUTINEPARMS di DB2. Il record contiene un valore che viene visualizzato nella colonna ORDINAL di SYSCAT.ROUTINEPARMS. Tale valore corrisponde a quello visualizzato nella colonna ORDINAL della vista DB2GSE.ST_GEOCODER_PARAMETERS.</p>
PARAMETER_NAME	VARCHAR(128)	Sì	<p>Nome del parametro. Se durante la creazione della funzione a cui appartiene il parametro non è stato indicato alcun nome, la colonna PARAMETER_NAME non conterrà alcun valore.</p> <p>Il contenuto della colonna PARAMETER_NAME viene richiamato dal catalogo del DB2.</p>
TYPE_SCHEMA	VARCHAR(128)	No	<p>Nome dello schema a cui appartiene il parametro. Il nome viene richiamato dal catalogo del DB2.</p>
TYPE_NAME	VARCHAR(128)	No	<p>Nome del tipo di dati dei valori assegnati al parametro. Il nome viene richiamato dal catalogo del DB2.</p>
PARAMETER_DEFAULT	VARCHAR(2048)	Sì	<p>Il valore predefinito da assegnare al parametro. DB2 considererà questo valore come una espressione SQL. Se il valore è racchiuso tra apici, viene trasmesso al geocoder come stringa. In caso contrario, l'espressione SQL determina il tipo di dati del parametro che verrà trasmesso al geocoder. Se la colonna PARAMETER_DEFAULT contiene un valore nullo, al geocoder viene trasmesso un valore nullo.</p> <p>E' possibile che il valore predefinito abbia un valore corrispondente nella vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS, e nell'input della procedura memorizzata ST_run_geocoding. Se uno dei valori è diverso dal valore predefinito, il nuovo valore sostituirà il valore predefinito.</p>
DESCRIPTION	VARCHAR(256)	Sì	<p>Descrizione del parametro che indica la relativa applicazione.</p>

Vista del catalogo DB2GSE.ST_GEOCODERS

Una volta abilitato un database per l'esecuzione delle operazioni spaziali, il geocoder fornito, DB2GSE_USA_GEOCODER, viene registrato nel catalogo DB2 Spatial Extender. Se si desidera mettere a disposizione degli utenti altri geocoder, sarà necessario registrarli. Per ottenere informazioni sui geocoder registrati, interrogare la vista del catalogo DB2GSE.ST_GEOCODERS. La seguente tabella contiene una descrizione delle colonne contenute nella vista.

Per ulteriori informazioni sui parametri dei geocoder, eseguire un'interrogazione la vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS di DB2 Spatial Extender e la vista SYSCAT.ROUTINEPARMS di DB2. Per informazioni sulle funzioni utilizzate come geocoder, eseguire un'interrogazione della vista del catalogo SYSCAT.ROUTINES di DB2.

Tabella 34. Colonne nella vista del catalogo DB2GSE.ST_GEOCODERS

Nome	Tipo di dati	Annullabile	Contenuto
GEOCODER_NAME	VARCHAR(128)	No	Nome del geocoder. Il nome è univoco nel database.
FUNCTION_SCHEMA	VARCHAR(128)	No	Nome dello schema a cui appartiene la funzione utilizzata come geocoder.
FUNCTION_NAME	VARCHAR(128)	No	Nome generico della funzione utilizzata come geocoder.
SPECIFIC_NAME	VARCHAR(128)	No	Nome specifico della funzione utilizzata come geocoder. I valori combinati di FUNCTION_SCHEMA e SPECIFIC_NAME identificano unicamente la funzione utilizzata come geocoder.
RETURN_TYPE_SCHEMA	VARCHAR(128)	No	Nome dello schema a cui appartiene il tipo di dati dell'output del geocoder. Il nome viene richiamato dal catalogo del DB2.
RETURN_TYPE_NAME	VARCHAR(128)	No	Nome generico del tipo di dati dell'output del geocoder. Il nome viene richiamato dal catalogo del DB2.
VENDOR	VARCHAR(256)	Sì	Nome del fornitore che ha creato il geocoder.
DESCRIPTION	VARCHAR(256)	Sì	Descrizione del geocoder che indica la propria applicazione.

Vista del catalogo DB2GSE.ST_GEOCODING

Durante l'impostazione delle operazioni di geocoding, i valori personalizzati assegnati vengono registrati automaticamente nel catalogo di DB2 Spatial Extender. Per richiamare tali impostazioni, eseguire un'interrogazione delle viste del catalogo DB2GSE.ST_GEOCODING e DB2GSE.ST_GEOCODING_PARAMETERS. La vista del catalogo DB2GSE.ST_GEOCODING, descritta nella seguente tabella, contiene i particolari relativi a tutte le impostazioni; ad esempio il numero di record elaborato da un geocoder prima di ciascun commit. La vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS contiene le impostazioni relative a ciascun geocoder. Ad esempio, le impostazioni relative al geocoder fornito, DB2GSE_USA_GEOCODER, comprendono il livello di corrispondenza minimo in base al quale l'indirizzo dato come input e gli indirizzi reali devono combaciare

vista del catalogo DB2GSE.ST_GEOCODING

affinché il geocoder esegua il geocoding dell'input. Questo requisito, definito *livello minimo di corrispondenza*, è registrato nella vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS.

Tabella 35. Colonne nella vista del catalogo DB2GSE.ST_GEOCODING

Nome	Tipo di dati	Annullabile	Contenuto
TABLE_SCHEMA	VARCHAR(128)	No	Nome dello schema che contiene la tabella a cui appartiene la colonna indicata nella colonna COLUMN_NAME.
TABLE_NAME	VARCHAR(128)	No	Nome generico della tabella che contiene la colonna indicata nella colonna COLUMN_NAME.
COLUMN_NAME	VARCHAR(128)	No	Nome della colonna spaziale in cui inserire i dati in base alle specifiche riportate nella vista del catalogo. I valori combinati nelle colonne TABLE_SCHEMA, TABLE_NAME e COLUMN_NAME identificano unicamente questa colonna spaziale.
GEOCODER_NAME	VARCHAR(128)	No	Nome del geocoder che verrà utilizzato per generare i dati da inserire nella colonna spaziale indicata nella colonna COLUMN_NAME. Ad una colonna spaziale può essere assegnato un solo geocoder.
MODE	VARCHAR(128)	No	Modalità di esecuzione del geocoding: BATCH E' abilitata solo l'esecuzione batch del geocoder. AUTO L'esecuzione automatica del geocoder è impostata e attiva. INVALIDO E' stata rilevata una inconsistenza tra le tabelle del catalogo spaziali; la richiesta di esecuzione del geocoding non è valida.
SOURCE_COLUMNS	VARCHAR(10000)	Sì	Nomi delle colonne della tabella impostate per l'esecuzione automatica del geocoder. Ogni qual volta le colonne verranno aggiornate, un trigger richiederà al geocoder di eseguire il geocoding dei dati aggiornati.
WHERE_CLAUSE	VARCHAR(10000)	Sì	Condizione di ricerca contenuta nella clausola WHERE. Tale condizione indica che se il geocoder viene eseguito in modalità batch, viene eseguito il geocoding solo dei dati contenuti nella serie di record indicata.
COMMIT_COUNT	INTEGER	Sì	Numero di righe da elaborare durante l'esecuzione del geocoder in modalità batch prima dell'esecuzione di un commit. Se il valore nella colonna COMMIT_COUNT è 0 (zero) o è nullo, non verrà eseguito alcun commit.

Vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS

Durante l'impostazione delle operazioni di geocoding per un determinato geocoder, le caratteristiche specifiche del geocoder vengono registrate automaticamente nel catalogo di Spatial Extender. Ad esempio, una delle operazioni tipiche del geocoder DB2GSE_USA_GEOCODER, consiste nel

vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS

paragonare gli indirizzi forniti come input ai dati di riferimento, quindi eseguire il geocoding degli indirizzi che corrispondono ai dati in base al livello di corrispondenza indicato o a un livello superiore. Durante l'impostazione delle operazioni per il geocoder, viene indicato il valore di corrispondenza, definito *livello minimo di corrispondenza* e viene registrato nel catalogo.

Per rilevare le caratteristiche specifiche di un geocoder rispetto alle impostazioni relative alle operazioni di geocoding, interrogare la vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS. La vista viene descritta nella seguente tabella.

La vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS, contiene alcuni valori predefiniti per l'impostazione delle operazioni di geocoding. I valori indicati nella vista DB2GSE.ST_GEOCODING_PARAMETERS sostituiranno i valori predefiniti.

Tabella 36. Colonne nella vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS

Nome	Tipo di dati	Annullabile	Contenuto
TABLE_SCHEMA	VARCHAR(128)	No	Nome dello schema che contiene la tabella a cui appartiene la colonna indicata nella colonna COLUMN_NAME.
TABLE_NAME	VARCHAR(128)	No	Nome della tabella che contiene la colonna spaziale.
COLUMN_NAME	VARCHAR(128)	No	Nome della colonna spaziale in cui inserire i dati in base alle specifiche riportate nella vista del catalogo. I valori combinati nelle colonne TABLE_SCHEMA, TABLE_NAME e COLUMN_NAME identificano unicamente questa colonna spaziale.
ORDINAL	SMALLINT	No	Posizione del parametro (ovvero il parametro indicato nella colonna PARAMETER_NAME) nella firma della funzione utilizzata come geocoder per la colonna COLUMN_NAME. Le informazioni relative a questo parametro sono disponibili anche in un record nella vista del catalogo SYSCAT.ROUTINEPARMS di DB2. Il record contiene un valore che viene visualizzato nella colonna ORDINAL di SYSCAT.ROUTINEPARMS. Tale valore corrisponde a quello visualizzato nella colonna ORDINAL della vista DB2GSE.ST_GEOCODING_PARAMETERS.
PARAMETER_NAME	VARCHAR(128)	Sì	Nome del parametro nella definizione del geocoder. Se durante la definizione del geocoder non è stato specificato alcun nome, il valore PARAMETER_NAME sarà nullo. Il contenuto della colonna PARAMETER_NAME viene richiamato dal catalogo del DB2.

vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS

Tabella 36. Colonne nella vista del catalogo DB2GSE.ST_GEOCODING_PARAMETERS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
PARAMETER_VALUE	VARCHAR(2048)	Sì	<p>Il valore assegnato al parametro. DB2 considererà questo valore come una espressione SQL. Se il valore è racchiuso tra apici, viene trasmesso al geocoder come stringa. In caso contrario, l'espressione SQL determina il tipo di dati del parametro che verrà trasmesso al geocoder. Se la colonna PARAMETER_VALUE contiene un valore nullo, al geocoder viene trasmesso un valore nullo.</p> <p>La colonna PARAMETER_VALUE corrisponde alla colonna PARAMETER_DEFAULT nella vista del catalogo DB2GSE.ST_GEOCODER_PARAMETERS. Se la colonna PARAMETER_VALUE contiene un valore, tale valore sostituirà il valore predefinito della colonna PARAMETER_DEFAULT. Se la colonna PARAMETER_VALUE non contiene alcun valore, viene utilizzato il valore predefinito.</p>

Vista del catalogo DB2GSE.ST_SIZINGS

Utilizzare la vista del catalogo DB2GSE.ST_SIZINGS per richiamare:

- Tutte le variabili supportate da Spatial Extender; ad esempio, *nome del sistema di coordinate*, *nome del geocoder* e le variabili a cui è possibile assegnare rappresentazioni di testo note di dati spaziali.
- La lunghezza massima consentita, se nota, dei valori assegnati a queste variabili (ad esempio, la lunghezza massima consentita per i nomi dei sistemi di coordinate, i nomi dei geocoder e le rappresentazioni di testo note dei dati spaziali).

La seguente tabella contiene una descrizione delle colonne contenute nella vista.

Tabella 37. Colonne nella vista del catalogo DB2GSE.ST_SIZINGS

Nome	Tipo di dati	Annullabile	Contenuto
VARIABLE_NAME	VARCHAR(128)	No	Termine che indica una variabile. Il termine è univoco nel database.
SUPPORTED_VALUE	INTEGER	Sì	<p>Lunghezza minima consentita per i valori assegnati alla variabile contenuta nella colonna VARIABLE_NAME. I valori che è possibile inserire nella colonna SUPPORTED_VALUE sono:</p> <p>Un valore numerico diverso da 0 Lunghezza massima consentita dei valori assegnati a questa variabile.</p> <p>0 E' consentito qualsiasi valore oppure non è possibile determinare la lunghezza consentita.</p> <p>NULLO Spatial Extender non supporta questa variabile.</p>
DESCRIPTION	VARCHAR(128)	Sì	Descrizione della variabile.

Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS

Per ottenere informazioni sui sistemi di riferimento spaziali registrati, interrogare la vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS. Spatial Extender registra automaticamente i sistemi di riferimento spaziali nel catalogo di Spatial Extender quando: times:

- Si abilita un database per le operazioni spaziali, cinque sistemi di riferimento spaziali predefiniti e 318 sistemi di riferimento spaziali geodetici. Per ulteriori informazioni, consultare le sezioni “Come stabilire se utilizzare il sistema di riferimento spaziale predefinito o crearne uno nuovo” a pagina 69 e “Datum supportati da DB2 Geodetic Extender” a pagina 211.
- Quando gli utenti creano altri sistemi di riferimento spaziali.

Per ottenere un valore completo dalla vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, bisogna tener presente che a ciascun sistema di riferimento spaziale è associato un sistema di coordinate. Il sistema di riferimento spaziale è stato ideato in parte per convertire le coordinate provenienti dal sistema di coordinate in valori che DB2 può elaborare con la massima efficienza, e in parte per definire l'estensione massima dello spazio a cui le coordinate possono riferirsi.

Per rilevare il nome e il tipo del sistema di coordinate associato al sistema di riferimento spaziale fornito, interrogare le colonne COORDSYS_NAME e COORDSYS_TYPE della vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS. Per ulteriori informazioni sul sistema di coordinate, interrogare la vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS.

Tabella 38. Colonne nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS

Nome	Tipo di dati	Annullabile	Contenuto
SRS_NAME	VARCHAR(128)	No	Nome del sistema di riferimento spaziale. Il nome è univoco nel database.
SRS_ID	INTEGER	No	Identificativo numerico del sistema di riferimento spaziale. Ciascun sistema di riferimento spaziale dispone di un identificativo numerico univoco. I valori SRS_ID dei sistemi di riferimento spaziali geodetici sono compresi tra 2000000000 e 2000001000. Le funzioni spaziali specificano i sistemi di riferimento spaziali mediante gli identificativi numerali anziché i nomi.
X_OFFSET	DOUBLE	No	Offset da sottrarre da tutte le coordinate X di una forma geometrica. La sottrazione viene eseguita durante la conversione delle coordinate geometriche in valori che DB2 è in grado di elaborare con la massima efficienza. La fase successiva prevede la moltiplicazione della figura ottenuta dalla sottrazione per il fattore di scala contenuto nella colonna X_SCALE.
X_SCALE	DOUBLE	No	Fattore di scala per il quale moltiplicare la figura ottenuta dalla sottrazione di un valore di scarto da una coordinata X. Il fattore è identico a quello contenuto nella colonna Y_SCALE.

Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS

Tabella 38. Colonne nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
Y_OFFSET	DOUBLE	No	Offset da sottrarre da tutte le coordinate Y di una forma geometrica. La sottrazione viene eseguita durante la conversione delle coordinate geometriche in valori che DB2 è in grado di elaborare con la massima efficienza. La fase successiva prevede la moltiplicazione della figura ottenuta dalla sottrazione per il fattore di scala contenuto nella colonna Y_SCALE.
Y_SCALE	DOUBLE	No	Fattore di scala per il quale moltiplicare la figura ottenuta dalla sottrazione di un valore di scarto da una coordinata Y. Il fattore è identico a quello contenuto nella colonna X_SCALE.
Z_OFFSET	DOUBLE	No	Offset da sottrarre da tutte le coordinate Z di una forma geometrica. La sottrazione viene eseguita durante la conversione delle coordinate geometriche in valori che DB2 è in grado di elaborare con la massima efficienza. La fase successiva prevede la moltiplicazione della figura ottenuta dalla sottrazione per il fattore di scala contenuto nella colonna Z_SCALE.
Z_SCALE	DOUBLE	No	Fattore di scala per il quale moltiplicare la figura ottenuta dalla sottrazione di un valore di scarto da una coordinata Z.
M_OFFSET	DOUBLE	No	Offset da sottrarre da tutte le misure di una forma geometrica. La sottrazione viene eseguita durante la conversione delle misure in valori che DB2 è in grado di elaborare con la massima efficienza. La fase successiva prevede la moltiplicazione della figura ottenuta dalla sottrazione per il fattore di scala contenuto nella colonna M_SCALE.
M_SCALE	DOUBLE	No	Fattore di scala per il quale moltiplicare la figura ottenuta dalla sottrazione di un valore di scarto da una misura.
MIN_X	DOUBLE	No	Valore minimo consentito per le coordinate X nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne X_OFFSET e X_SCALE.
MAX_X	DOUBLE	No	Valore massimo consentito per le coordinate X nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne X_OFFSET e X_SCALE.
MIN_Y	DOUBLE	No	Valore minimo consentito per le coordinate Y nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne Y_OFFSET e Y_SCALE.
MAX_Y	DOUBLE	No	Valore massimo consentito per le coordinate Y nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne Y_OFFSET e Y_SCALE.

Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS

Tabella 38. Colonne nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS (Continua)

Nome	Tipo di dati	Annullabile	Contenuto
MIN_Z	DOUBLE	No	Valore minimo consentito per le coordinate Z nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne Z_OFFSET e Z_SCALE.
MAX_Z	DOUBLE	No	Valore massimo consentito per le coordinate Z nelle forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne Z_OFFSET e Z_SCALE.
MIN_M	DOUBLE	No	Valore minimo consentito per le misure che è possibile memorizzare con le forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne M_OFFSET e M_SCALE.
MAX_M	DOUBLE	No	Valore massimo consentito per le misure che è possibile memorizzare con le forme geometriche a cui viene applicato questo sistema di riferimento spaziale. Questo valore deriva dai valori contenuti nelle colonne M_OFFSET e M_SCALE.
COORDSYS_NAME	VARCHAR(128)	No	Nome che identifica il sistema di coordinate su cui si basa questo sistema di riferimento spaziale.
COORDSYS_TYPE	VARCHAR(128)	No	Il tipo di sistema di coordinate su cui si basa questo sistema di riferimento spaziale.
ORGANIZATION	VARCHAR(128)	Sì	Nome dell'organizzazione (ad esempio, un testo standard) che definisce il sistema di coordinate su cui si basa questo sistema spaziale di riferimento. ORGANIZATION è nullo se ORGANIZATION_COORSYS_ID è nullo.
ORGANIZATION_COORSYS_ID	INTEGER	Sì	Nome dell'organizzazione (ad esempio, un testo standard) che definisce il sistema di coordinate su cui si basa questo sistema spaziale di riferimento. ORGANIZATION_COORSYS_ID è nullo se ORGANIZATION è nullo.
DEFINITION	VARCHAR(2048)	No	Rappresentazione WKT (well-known text) della definizione del sistema di coordinate.
DESCRIPTION	VARCHAR(256)	Sì	Descrizione del sistema di riferimento spaziale.

Argomenti correlati:

- "Sistemi di riferimento spaziali" a pagina 68

Attività correlate:

- "Creazione di un sistema di riferimento spaziale" a pagina 75

Vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE

Alcune funzioni spaziali, accettano o restituiscono i valori che denotano una distanza specifica. In alcuni casi, è possibile scegliere l'unità di misura in cui esprimere la distanza. Ad esempio, ST_Distance restituisce la distanza minima tra due determinate forme geometriche. E' possibile richiedere a ST_Distance di

vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE

restituire la distanza in miglia o in metri. Per individuare le unità di misura disponibili, consultare la vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Tabella 39. Colonne nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE

Nome	Tipo di dati	Annullabile	Contenuto
UNIT_NAME	VARCHAR(128)	No	Nome dell'unità di misura. Il nome è univoco nel database.
UNIT_TYPE	VARCHAR(128)	No	Tipo di unità di misura. I valori possibili sono: LINEAR L'unità di misura è lineare. ANGULAR L'unità di misura è angolare.
CONVERSION_FACTOR	DOUBLE	No	Valore numerico utilizzato per convertire l'unità di misura nella corrispondente unità di base. L'unità di base per le unità di misura lineari è il METRO; per le unità di misura angolari è il RAGGIO. L'unità di base ha un fattore di conversione di 1,0.
DESCRIPTION	VARCHAR(256)	Sì	Descrizione dell'unità di misura.

Capitolo 22. Funzioni spaziali: categorie e uso

In questo capitolo vengono introdotte le funzioni spaziali, suddivise in categorie.

Funzioni spaziali

DB2[®] Spatial Extender fornisce funzioni che:

- Convertono le forme geometriche in e da diversi formati di scambio dati. Tali funzioni sono definite *funzioni di costruzione*.
- Confrontano le forme geometriche per rilevare confini, intersezioni ed altre informazioni. Tali funzioni sono definite *funzioni di confronto*.
- Restituiscono informazioni sulle proprietà delle forme geometriche, ad esempio coordinate e misure interne alle forme, relazioni tra le forme, confini ed altre informazioni.
- Generano nuove forme geometriche da forme geometriche esistenti.
- Misurano la distanza più breve tra i punti di una forma geometrica.
- Forniscono informazioni sui parametri di indice.
- Forniscono proiezioni e conversioni tra diversi sistemi di coordinate.

Argomenti correlati:

- “Funzioni che restituiscono informazioni sulla distanza” a pagina 332
- “Funzioni che restituiscono informazioni sull’indice” a pagina 333
- “Conversioni tra sistemi di coordinate” a pagina 333

Riferimenti correlati:

- “Esempi delle operazioni eseguite dalle funzioni spaziali” a pagina 121
- “Funzioni che restituiscono di informazioni sulle proprietà delle forme geometriche” a pagina 318
- “Funzioni che confrontano le funzioni geografiche” a pagina 304
- “Funzioni che generano nuove forme geometriche da forme geometriche esistenti” a pagina 325
- “Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati” a pagina 295

Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati

DB2 Spatial Extender fornisce funzioni spaziali in grado di convertire forme geometriche in e da i seguenti formati di scambio dati:

- Rappresentazione WKT (Well-known text)
- Rappresentazione WKB (Well-known binary)
- Rappresentazione shape ESRI
- Rappresentazione GML (Geography Markup Language)

Le funzioni per la creazione di forme geometriche da questi formati sono chiamate *funzioni di costruzione*.

Argomenti correlati:

- “Panoramica sulle funzioni di costruzione” a pagina 296
- “Conversione in rappresentazioni WKT (well-known text)” a pagina 300
- “conversione in rappresentazioni WKB (well-known binary)” a pagina 301
- “Conversione in rappresentazioni shape ESRI” a pagina 302
- “Conversione in rappresentazioni GML (Geography Markup Language)” a pagina 303

Riferimenti correlati:

- “Gruppi di trasformazione” a pagina 509

Panoramica sulle funzioni di costruzione

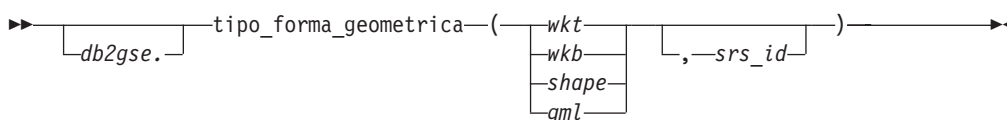
Le funzioni di costruzione hanno lo stesso nome dei tipi di dati geometrici della colonna in cui i dati verranno inseriti. Queste funzioni operano costantemente su ciascun formato di scambio dati di input. Questa sezione contiene:

- L'SQL per richiamare le funzioni che operano su formati di scambio dati e il tipo di forma geometrica restituita dalle funzioni
- L'SQL per richiamare una funzione che consente di creare punti da coordinate X e Y, e il tipo di forma geometrica restituito da questa funzione
- Esempi di codici e serie di risultati

Funzioni che operano su formati di scambio dati

Questa sezione riporta la sintassi necessaria per richiamare le funzioni che operano sui formati di scambio dati, descrive i parametri di input per le funzioni ed identifica il tipo di forme geometriche restituite dalle funzioni.

Sintassi:



Parametri ed altri elementi della sintassi:

db2gse Nome dello schema a cui appartengono i tipi di dati spaziali forniti da DB2® Spatial Extender.

tipo_forma_geometrica

Una delle seguenti funzioni di costruzione:

- ST_Point
- ST_LineString
- ST_Polygon
- ST_MultiPoint
- ST_MultiLineString
- ST_MultiPolygon
- ST_GeomCollection
- ST_Geometry

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della forma geometrica.

Funzioni spaziali

coordinata_z

Un valore di tipo DOUBLE che specifica la coordinata Z del punto ottenuto come risultato.

Se il parametro *coordinata_z* viene omissso, il punto ottenuto come risultato non avrà coordinata Z. Per tale punto, il risultato di ST_Is3D è 0 (zero).

coordinata_m

Un valore di tipo DOUBLE che specifica la coordinata M del punto ottenuto come risultato.

Se il parametro *coordinata_m* viene omissso, il punto ottenuto come risultato non avrà misure. Per tale punto, il risultato di ST_IsMeasured è 0 (zero).

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il punto ottenuto come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Point

Esempi

Questa sezione contiene esempi di codice utilizzato per richiamare funzioni di costruzione, per creare tabelle in cui inserite l'output della funzione di costruzione, per richiamare l'output e lo stesso output.

L'esempio seguente consente di inserire una riga nella tabella SAMPLE_GEOMETRY con ID 100 e un valore punto con coordinata X di 30, coordinata Y di 40 e nel sistema di riferimento spaziale 1 utilizzando la rappresentazione di coordinate WKT. Quindi inserisce un'altra riga con ID 200 e un valore linea con le coordinate indicate.

```
CREATE TABLE sample_geometry (id INT, geom db2gse.ST_Geometry);

INSERT INTO sample_geometry(id, geom)
VALUES(100,db2gse.ST_Geometry('point(30 40)', 1));

INSERT INTO sample_geometry(id, geom)
VALUES(200,db2gse.ST_Geometry('linestring(50 50, 100 100)', 1));

SELECT id, TYPE_NAME(geom) FROM sample_geometry

ID      2
-----
100 "ST_POINT"
200 "ST_LINESTRING"
```

Se la colonna spaziale può contenere solo valori ST_Point, è possibile utilizzare il seguente esempio per inserire due punti. Inserendo una linea o qualsiasi altro tipo diverso da un punto, si verificherà un errore SQL. Il primo inserimento consente di creare un punto dalla rappresentazione WKT. Il secondo inserimento consente di creare un punto da valori di coordinate numerici. I valori di input possono anche essere selezionati dalle colonne di una tabella esistente.

```
CREATE TABLE sample_points (id INT, geom db2gse.ST_Point);

INSERT INTO sample_points(id, geom)
VALUES(100,db2gse.ST_Point('point(30 40)', 1));

INSERT INTO sample_points(id, geom)
VALUES(101,db2gse.ST_Point(50, 50, 1));

SELECT id, TYPE_NAME(geom) FROM sample_geometry

ID      2
-----
100 "ST_POINT"
101 "ST_POINT"
```

Nell'esempio seguente viene utilizzato Embedded SQL e si presuppone che l'applicazione inserisca nelle aree dei dati i valori appropriati.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id = 0;
    SQL TYPE IS CLOB(10000) wkt_buffer;
    SQL TYPE IS CLOB(10000) gml_buffer;
    SQL TYPE IS BLOB(10000) wkb_buffer;
    SQL TYPE IS BLOB(10000) shape_buffer;
EXEC SQL END DECLARE SECTION;

// * Inserire in questo punto la logica dell'applicazione
// per la lettura nelle aree di rispetto */

EXEC SQL INSERT INTO sample_geometry(id, geom)
VALUES(:id, db2gse.ST_Geometry(:wkt_buffer,1));

EXEC SQL INSERT INTO sample_geometry(id, geom)
VALUES:id, db2gse.ST_Geometry(:wkb_buffer,1));

EXEC SQL INSERT INTO sample_geometry(id, geom)
VALUES(:id, db2gse.ST_Geometry(:gml_buffer,1));

EXEC SQL INSERT INTO sample_geometry(id, geom)
VALUES(:id, db2gse.ST_Geometry(:shape_buffer,1));
```

Il codice Java™ del seguente esempio, utilizza JDBC per inserire punti e mediante i valori di coordinate X e Y numerici, e usa la rappresentazione WTK per specificare le forme geometriche.

```
String ins1 = "INSERT into sample_geometry (id, geom)
VALUES(?, db2gse.ST_PointFromText(CAST( ?
as VARCHAR(128)), 1))";
PreparedStatement pstmt = con.prepareStatement(ins1);
pstmt.setInt(1, 100); // id value
pstmt.setString(2, "point(32.4 50.7)"); // wkt value
int rc = pstmt.executeUpdate();

String ins2 = "INSERT into sample_geometry (id, geom)
VALUES(?, db2gse.ST_Point(CAST( ? as double),
CAST(? as double), 1))";
pstmt = con.prepareStatement(ins2);
pstmt.setInt(1, 200); // id value
pstmt.setDouble(2, 40.3); // lat
pstmt.setDouble(3, -72.5); // long
rc = pstmt.executeUpdate();
```

Riferimenti correlati:

- “Gruppi di trasformazione” a pagina 509

Conversione in rappresentazioni WKT (well-known text)

Le rappresentazioni di testo sono valori CLOB che rappresentano stringhe di caratteri ASCII e consentono di trasformare le forme geometriche in formato testo ASCII.

La funzione **ST_AsText** consente di convertire una forma geometrica memorizzata in una tabella in una stringa WKT. Nel seguente esempio, per selezionare i valori inseriti precedentemente nella tabella `SAMPLE_GEOMETRY`, viene utilizzata una semplice query dalla riga comandi.

```
SELECT id, VARCHAR(db2gse.ST_AsText(geom), 50) AS WKTGEOM
FROM sample_geometry;
```

```
ID    WKTGEOM
-----
100   POINT ( 30.00000000 40.00000000)
200   LINESTRING ( 50.00000000 50.00000000, 100.00000000 100.00000000)
```

Nel seguente esempio, per selezionare i valori inseriti precedentemente nella tabella `SAMPLE_GEOMETRY`, viene utilizzato Embedded SQL.

```
EXEC SQL BEGIN DECLARE SECTION;
sqlint32 id = 0;
SQL TYPE IS CLOB(10000) wkt_buffer;
short wkt_buffer_ind = -1;
EXEC SQL END DECLARE SECTION;

EXEC SQL
SELECT id, db2gse.ST_AsText(geom)
INTO :id, :wkt_buffer :wkt_buffer_ind
FROM sample_geometry
WHERE id = 100;
```

In alternativa, per convertire le forme geometriche nella corrispondente rappresentazione WKT, è possibile utilizzare il gruppo di trasformazione `ST_WellKnownText`. Il seguente codice di esempio mostra come utilizzare il gruppo di trasformazione.

```
EXEC SQL BEGIN DECLARE SECTION;
sqlint32 id = 0;
SQL TYPE IS CLOB(10000) wkt_buffer;
short wkt_buffer_ind = -1;
EXEC SQL END DECLARE SECTION;

EXEC SQL
SET CURRENT DEFAULT TRANSFORM GROUP = ST_WellKnownText;

EXEC SQL
SELECT id, geom
INTO :id, :wkt_buffer :wkt_buffer_ind
FROM sample_geometry
WHERE id = 100;
```

L'istruzione `SELECT` non utilizza alcuna funzione spaziale per convertire la forma geometrica.

DB2[®] Spatial Extender fornisce altre funzioni, oltre a quelle trattate in questa sezione, per la conversione delle forme geometriche in e da rappresentazioni WKT. DB2 Spatial Extender fornisce tali funzioni per uniformarsi alla specifica "Simple Features for SQL" di OGC e agli standard ISO SQL/MM Part 3: Spatial standard. Queste funzioni sono:

- ST_WKTTToSQL
- ST_GeomFromText
- ST_GeomCollFromTxt
- ST_PointFromText
- ST_LineFromText
- ST_PolyFromText
- ST_MPointFromText
- ST_MLineFromText
- ST_MPolyFromText

Riferimenti correlati:

- “Gruppi di trasformazione” a pagina 509

conversione in rappresentazioni WKB (well-known binary)

Le rappresentazioni WKB consistono in strutture di dati binari che devono essere formate da valori BLOB. Tali valori BLOB rappresentano strutture di dati binari che dovranno essere gestite da un programma applicativo scritto in un linguaggio di programmazione supportato da DB2® e per il quale DB2 disponga di linguaggio appropriato.

La funzione **ST_AsBinary** converte una forma geometrica memorizzata in una tabella nella rappresentazione WKB che è possibile recuperare in una variabile BLOB nella memoria del programma. Nel seguente esempio, per selezionare i valori inseriti precedentemente nella tabella SAMPLE_GEOMETRY, viene utilizzato Embedded SQL.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id = 0;
    SQL TYPE IS BLOB(10000) wkb_buffer;
    short wkb_buffer_ind = -1;
EXEC SQL END DECLARE SECTION;

EXEC SQL
    SELECT id, db2gse.ST_AsBinary(geom)
    INTO :id, :wkb_buffer :wkb_buffer_ind
    FROM sample_geometry
    WHERE id = 200;
```

In alternativa, per convertire le forme geometriche nella corrispondente rappresentazione WKB, è possibile utilizzare il gruppo di trasformazione ST_WellKnownBinary. Il seguente codice di esempio mostra come utilizzare il gruppo di trasformazione.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id = 0;
    SQL TYPE IS BLOB(10000) wkb_buffer;
    short wkb_buffer_ind = -1;
EXEC SQL END DECLARE SECTION;

EXEC SQL
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_WellKnownBinary;

EXEC SQL
    SELECT id, geom
    INTO :id, :wkb_buffer :wkb_buffer_ind
    FROM sample_geometry
    WHERE id = 200;
```

Funzioni spaziali

L'istruzione SELECT non utilizza alcuna funzione spaziale per convertire la forma geometrica.

Esistono altre funzioni, oltre a quelle trattate in questa sezione, per la conversione delle forme geometriche in e da rappresentazioni WKB. DB2 Spatial Extender fornisce tali funzioni per uniformarsi alla specifica "Simple Features for SQL" di OGC e agli standard ISO SQL/MM Part 3: Spatial standard. Queste funzioni sono:

- **ST_WKBToSQL**
- **ST_GeomFromWKB**
- **ST_GeomCollFromWKB**
- **ST_PointFromWKB**
- **ST_LineFromWKB**
- **ST_PolyFromWKB**
- **ST_MPointFromWKB**
- **ST_MLineFromWKB**
- **ST_MPolyFromWKB**

Riferimenti correlati:

- "Gruppi di trasformazione" a pagina 509

Conversione in rappresentazioni shape ESRI

La rappresentazione shape ESRI consiste in strutture di dati binari che dovranno essere gestite da un programma applicativo scritto in un linguaggio supportato.

La funzione **ST_AsShape** converte una forma geometrica memorizzata in una tabella nella rappresentazione shape ESRI che è possibile recuperare in una variabile BLOB nella memoria del programma. Nel seguente esempio, per selezionare i valori inseriti precedentemente nella tabella SAMPLE_GEOMETRY, viene utilizzato Embedded SQL.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id;
    SQL TYPE IS BLOB(10000) shape_buffer;
EXEC SQL END DECLARE SECTION;
```

```
EXEC SQL
    SELECT id, db2gse.ST_AsShape(geom)
    INTO :id, :shape_buffer
    FROM sample_geometry;
```

In alternativa, per convertire le forme geometriche nella corrispondente rappresentazione shape, è possibile utilizzare il gruppo di trasformazione ST_Shape. Il seguente codice di esempio mostra come utilizzare il gruppo di trasformazione.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id = 0;
    SQL TYPE IS BLOB(10000) shape_buffer;
    short shape_buffer_ind = -1;
EXEC SQL END DECLARE SECTION;

EXEC SQL
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_Shape;
```



```
EXEC SQL
  SELECT id, geom
  FROM sample_geometry
  WHERE id = 300;
```

L'istruzione SELECT non utilizza alcuna funzione spaziale per convertire la forma geometrica.

Riferimenti correlati:

- “Gruppi di trasformazione” a pagina 509

Conversione in rappresentazioni GML (Geography Markup Language)

Le rappresentazioni GML (Geography Markup Language) sono stringhe ASCII e consentono di trasformare le forme geometriche in formato testo ASCII.

La funzione **ST_AsGML** consente di convertire una forma geometrica memorizzata in una tabella in una stringa di testo GML. Il seguente esempio, consente di selezionare i valori inseriti precedentemente nella tabella SAMPLE_GEOMETRY. I risultati mostrati nell'esempio, sono stati formattati nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

```
SELECT id, VARCHAR(db2gse.ST_AsGML(geom), 500) AS GMLGEOM
FROM sample_geometry;
```

ID	GMLGEOM
100	<gml:Point srsName="EPSG:4269"> <gml:coord><gml:X>30</gml:X><gml:Y>40</gml:Y></gml:coord> </gml:Point>
200	<gml:LineString srsName="EPSG:4269"> <gml:coord><gml:X>50</gml:X><gml:Y>50</gml:Y></gml:coord> <gml:coord><gml:X>100</gml:X><gml:Y>100</gml:Y></gml:coord> </gml:LineString>

In alternativa, per convertire le forme geometriche nella corrispondente rappresentazione HTML, è possibile utilizzare il gruppo di trasformazione **ST_GML**.

```
SET CURRENT DEFAULT TRANSFORM GROUP = ST_GML
```

```
SELECT id, geom AS GMLGEOM
FROM sample_geometry;
```

ID	GMLGEOM
100	<gml:Point srsName="EPSG:4269"> <gml:coord><gml:X>30</gml:X><gml:Y>40</gml:Y></gml:coord> </gml:Point>
200	<gml:LineString srsName="EPSG:4269"> <gml:coord><gml:X>50</gml:X><gml:Y>50</gml:Y></gml:coord> <gml:coord><gml:X>100</gml:X><gml:Y>100</gml:Y></gml:coord> </gml:LineString>

L'istruzione SELECT non utilizza alcuna funzione spaziale per convertire la forma geometrica.

Riferimenti correlati:

- “Gruppi di trasformazione” a pagina 509

Funzioni che confrontano le funzioni geografiche

Alcune funzioni spaziali restituiscono informazioni sulle relazioni che intercorrono tra caratteristiche geografiche collegate tra loro. Altre funzioni, possono rilevare se due definizioni di coordinate o due sistemi di coordinate spaziali coincidono. In tutti i casi, le informazioni restituite sono ottenute da un confronto tra forme geometriche, tra definizioni di sistemi di coordinate o tra sistemi di riferimento spaziali. Le funzioni che forniscono tali informazioni sono definite *funzioni di confronto*.

Le funzioni di confronto sono:

- **ST_Contains e ST_Within.** Queste funzioni utilizzano entrambe due forme geometriche come input e determinano se lo spazio interno di una si interseca con lo spazio interno dell'altra.
- **ST_Intersects, ST_Crosses, ST_Overlaps e ST_Touches.** Queste funzioni restituiscono informazioni sulle intersezioni delle forme geometriche.
- **ST_EnvIntersects e ST_MBRIntersects.** Queste funzioni determinano se il rettangolo minore che racchiude una forma geometrica si interseca con il rettangolo minore che racchiude l'altra forma.
- **ST_Equals, ST_EqualCoordsys e ST_EqualsSRS.** Queste funzioni determinano se i due elementi da mettere a confronto sono identici.
- **ST_Relate.** Questa funzione determina se le forme geometriche messe a confronto soddisfano la condizione della stringa di matrice modello DE-9IM.
- **ST_Disjoint.** Questa funzione verifica che non vi siano intersezioni tra le forme geometriche.

Argomenti correlati:

- "Funzioni che confrontano forme geometriche alla stringa matrice di modello DE-9IM" a pagina 318
- "Panoramica delle funzioni di confronto" a pagina 304
- "Funzioni che controllano se una forma geometrica ne contiene un'altra" a pagina 306
- "Funzioni che controllano le intersezioni tra forme geometriche" a pagina 309
- "Funzioni che confrontano gli involucri delle forme geometriche" a pagina 315
- "Funzioni che controllano la similitudine di due elementi" a pagina 315
- "Funzioni che controllano la presenza di intersezioni tra due forme geometriche" a pagina 317

Panoramica delle funzioni di confronto

Le funzioni di confronto di DB2[®] Spatial Extender restituiscono valore 1 (uno) se il confronto soddisfa determinati criteri, valore 0 (zero) se nessun criterio viene soddisfatto o valore nullo se non risulta possibile eseguire il confronto. Non è possibile eseguire confronti se l'operazione di confronto non è stata definita per i parametri di input o se uno dei parametri ha valore nullo. I confronti *possono* essere eseguiti se ai parametri vengono assegnate forme geometriche aventi tipi di dati o dimensioni differenti.

Il *Dimensionally Extended 9 Intersection Model (DE-9IM)* è un approccio matematico che definisce i rapporti spaziali a coppie tra le forme geometriche o diversi tipi e dimensioni. Questo modello esprime i rapporti spaziali tra tutti i tipi di forme

geometriche come intersezioni a coppie dei rispettivi spazi interni, contorni e spazi esterni, considerando la dimensione delle intersezioni risultanti.

Le forme geometriche date a e b : $I(a)$, $B(a)$ ed $E(a)$ rappresentano rispettivamente lo spazio interno, il contorno e lo spazio esterno di a . $I(b)$, $B(b)$ ed $E(b)$ rappresentano lo spazio interno, il contorno e lo spazio esterno di b . Le intersezioni di $I(a)$, $B(a)$ ed $E(a)$ con $I(b)$, $B(b)$ ed $E(b)$ generano una matrice 3-per-3. Ciascuna intersezione può risultare in una forma geometrica di diverse dimensioni. Ad esempio, se l'intersezione dei contorni di due poligoni consiste in un punto e una linea, la funzione \dim restituisce una dimensione massima di 1.

La funzione \dim restituisce i valori -1 , 0 , 1 o 2 . Il valore -1 corrisponde all'impostazione null o $\dim(\text{null})$, restituita se non viene rilevata alcuna intersezione.

	Interno	Contorno	Esterno
Interno	$\dim(I(a) \cap I(b))$	$\dim(I(a) \cap B(b))$	$\dim(I(a) \cap E(b))$
Contorno	$\dim(B(a) \cap I(b))$	$\dim(B(a) \cap B(b))$	$\dim(B(a) \cap E(b))$
Esterno	$\dim(E(a) \cap I(b))$	$\dim(E(a) \cap B(b))$	$\dim(E(a) \cap E(b))$

I risultati ottenuti dalle funzioni di confronto possono essere interpretati o verificati paragonandoli ai risultati ottenuti da una funzione di confronto avente una matrice modello che rappresenta i valori accettabili per il DE-9IM.

La matrice modello conterrà i valori accettabili per ciascuna intersezione delle celle della matrice. I valori possibili sono:

- T** Intersezione obbligatoria; $\dim = 0, 1$ o 2 .
- F** Non si richiede intersezione; $\dim = -1$.
- *** Intersezione non rilevante; $\dim = -1, 0, 1$ o 2 .
- 0** Intersezione obbligatoria e le dimensioni dovranno essere 0 ; $\dim = 0$.
- 1** Intersezione obbligatoria e le dimensioni massime dovranno essere 1 ; $\dim = 1$.
- 2** Intersezione obbligatoria e le dimensioni massime dovranno essere 2 ; $\dim = 2$.

Ad esempio, la seguente matrice modello per la funzione ST_Within contiene i valori T, F e *.

Tabella 40. Matrice per ST_Within. Matrice modello della funzione ST_Within per combinazioni di forme geometriche.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Interno forma geometrica a	T	*	F
Contorno forma geometrica a	*	*	F
Esterno forma geometrica a	*	*	*

Funzioni spaziali

La funzione ST_Within restituisce valore 1 se gli spazi interni delle due forme geometriche si intersecano e se lo spazio interno o il contorno di *a* non si interseca con lo spazio esterno di *b*. Tutte le altre condizioni non influiscono sul risultato della funzione.

Ciascuna funzione dispone di almeno una matrice modello, ma alcune funzioni per descrivere i rapporti tra le diverse combinazioni di tipi geometrici, richiedono più matrici.

Il DE-9IM è stato sviluppato da Clementini e Felice, che hanno esteso alle dimensioni il modello 9 Intersection Model di Egenhofer ed Herring. Il DE-9IM è il risultato di una collaborazione di quattro autori (Clementini, Eliseo, Di Felice e van Osstrom) che hanno pubblicato il modello in "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," D. Abel and B.C. Ooi (Ed.), *Advances in Spatial Database—Third International Symposium. SSD '93*. LNCS 692. Pp. 277-295. Il modello 9 Intersection model di M. J. Egenhofer e J. Herring (Springer-Verlag Singapore [1993]) è stato pubblicato in "Categorizing binary topological relationships between regions, lines, and points in geographic databases," *Tech. Report, Department of Surveying Engineering, University of Maine, Orono, ME 1991*.

Elenco delle funzioni

Le funzioni di confronto sono:

- ST_Contains
- ST_Crosses
- ST_Disjoint
- ST_EnvIntersects
- ST_EqualCoordsys
- ST_Equals
- ST_EqualSRS
- ST_Intersects
- ST_MBRIntersects
- ST_Overlaps
- ST_Relate
- ST_Touches
- ST_Within

Funzioni che controllano se una forma geometrica ne contiene un'altra

ST_Contains e ST_Within utilizzano entrambe due forme geometriche come input e determinano se lo spazio interno di una si interseca con lo spazio interno dell'altra. In termini pratici, ST_Contains stabilisce se la prima forma geometrica data include la seconda forma (se la seconda forma è contenuta nella prima). ST_Within stabilisce se la prima forma geometrica si trova completamente all'interno della seconda (ovvero se la prima forma è contenuta nella seconda).

ST_Contains

ST_Contains restituisce valore 1 (uno) se la seconda forma geometrica è contenuta completamente nella prima forma. La funzione ST_Contains restituisce esattamente il risultato opposto della funzione ST_Within.

La Figura 44 mostra alcuni esempi di ST_Contains:

- Una forma geometrica multipunto contiene forme geometriche punto o un multipunto quando tutti i punti si trovano all'interno della prima forma geometrica.
- Una forma geometrica poligono, contiene una forma geometrica multipunto quando tutti i punti si trovano sul contorno del poligono o al suo interno.
- Una forma geometrica linea contiene forme geometriche punto, multipunto o linee, quando tutti i punti si trovano all'interno della prima forma geometrica.
- Una forma geometrica poligono, contiene forme geometriche punto, linea o poligono, quando la seconda forma geometrica si trova all'interno del poligono.

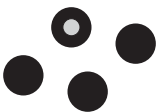

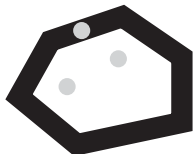




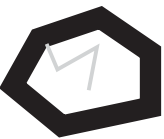

		
multipunto / punto	multipunto / multipunto	poligono / multipunto
		
linea / punto	linea / multipunto	linea / linea
		
poligono / punto	poligono / linea	poligono / poligono

Figura 44. ST_Contains. Le forme geometriche scure, rappresentano la forma geometrica a e le forme grigie la b. In tutti i casi la forma geometrica a contiene completamente la forma geometrica b.

La matrice modello della funzione ST_Contains specifica che gli spazi interni delle due forme forme geometriche devono intersecarsi e che gli spazi interni o i contorni delle forma secondaria (b) non devono intersecare lo spazio esterno della forma principale (a).L'asterisco (*) indica che la presenza di un'intersezione tra queste parti delle forme geometriche non ha alcuna conseguenza.

Tabella 41. Matrice per ST_Contains

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Interno forma geometrica a	T	*	*
Contorno forma geometrica a	*	*	*
Esterno forma geometrica a	F	F	*

ST_Within

ST_Within restituisce valore 1 (uno) se la prima forma geometrica è contenuta completamente nella seconda forma. Il risultato della funzione ST_Within è opposto al risultato della funzione ST_Contains.

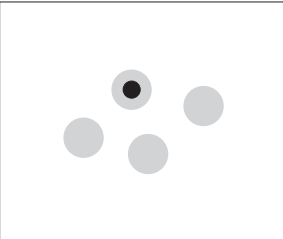
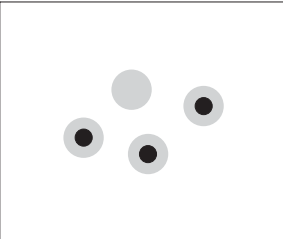
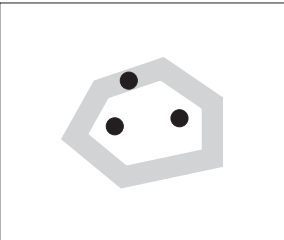
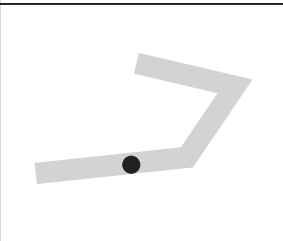
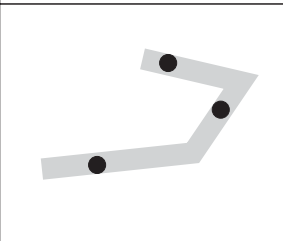
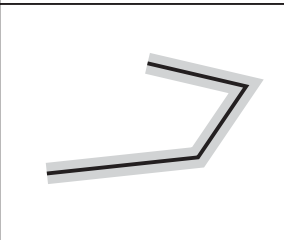
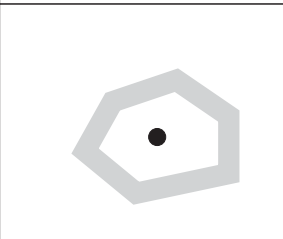
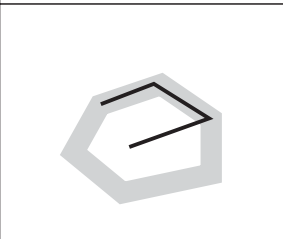
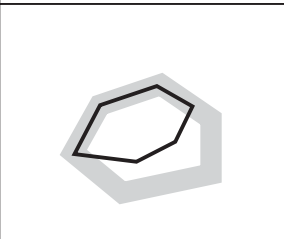
		
punto / multipunto	multipunto / multipunto	multipunto / poligono
		
punto / linea	multipunto / linea	linea / linea
		
punto / poligono	linea / poligono	poligono / poligono

Figura 45. ST_Within

La matrice modello della funzione ST_Within indica che gli spazi interni delle due forme geometriche devono intersecarsi e che lo spazio interno o il contorno della forma geometrica principale (a) non devono intersecare lo spazio esterno della forma secondaria (b).L'asterisco (*) indica che tutte le altre intersezioni non hanno alcuna conseguenza.

Tabella 42. Matrice per ST_Within

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Interno forma geometrica a	T	*	F
Contorno forma geometrica a	*	*	F
Esterno forma geometrica a	*	*	*

La Figura 45 a pagina 308 mostra alcuni esempi di ST_Within:

- Una forma geometrica a punti è interna ad una forma geometrica a multipunti quando il suo interno si interseca con uno dei punti della seconda forma geometrica.
- Una forma geometrica a multipunti è interna ad una forma geometrica a multipunti quando la parte interna di tutti i punti si interseca con la seconda forma geometrica.
- Una forma geometrica multipunto è interna alla geometrica poligono quando tutti i punti si trovano sul contorno del poligono o al suo interno.
- Una forma geometrica a punti è interna ad una forma geometrica a linee tutti i punti si trovano all'interno della seconda forma geometrica. Nella Figura 45 a pagina 308, il punto non è interno alla linea perché il suo interno non interseca la linea; tuttavia, la forma geometrica a multipunti è interna alla linea perché tutti i punti che la compongono si intersecano con l'interno della linea.
- Una forma geometrica a linee è interna ad altre forme geometriche a linee quando tutti i punti si intersecano con la seconda forma geometrica.
- Una forma geometrica a punti, non è interna a una forma geometrica poligono perché il suo interno non interseca il contorno o l'interno del poligono.
- Una forma geometrica a linee è interna ad un poligono quando tutti i punti che la compongono intersecano il contorno o l'interno del poligono.
- Una poligono è interno a un poligono quando tutti i punti che lo compongono intersecano il contorno o l'interno del poligono.

Funzioni che controllano le intersezioni tra forme geometriche

ST_Intersects, ST_Crosses, ST_Overlaps e ST_Touches sono tutte funzioni che rilevano eventuali intersezioni tra forme geometriche. La differenza consiste essenzialmente nelle parti che si intersecano:

- ST_Intersects controlla se due forme geometriche date incontrano una di quattro condizioni: intersezione degli spazi interni delle forme geometriche, dei contorni, intersezione del contorno della prima forma geometrica con lo spazio interno della seconda o intersezione dello spazio interno della prima forma geometrica con il contorno della seconda.
- ST_Crosses viene utilizzata per analizzare l'intersezione di forme geometriche di diverse dimensioni, con una eccezione: può analizzare anche l'intersezione di linee. In tutti i casi, l'intersezione stessa è considerata una forma geometrica; ST_Crosses richiede che questa forma geometrica sia più piccola delle forme geometriche intersecanti più grandi (oppure, se si tratta di linee, che il punto di intersezione abbia dimensioni minori della linea). Ad esempio, la dimensioni di una linea e un poligono sono rispettivamente 1 e 2. Se le due forme geometriche si intersecano, e l'intersezione è lineare (il percorso della linea intorno al poligono), l'intersezione stessa può essere considerata una forma geometrica.

Poiché la dimensione della linea (1) è minore della dimensione del poligono (2), ST_Crosses, dopo aver analizzato l'intersezione, restituirà valore 1.

- Le forme geometriche utilizzate come input da ST_Overlaps devono avere la stessa dimensione. ST_Overlaps richiede che tali forme geometriche si sovrappongano parzialmente, formando una nuova forma geometrica (l'area di sovrapposizione) avente la stessa dimensione delle forme.
- ST_Touches stabilisce se i contorni di due forme geometriche si intersecano o meno.

ST_Intersects

ST_Intersects restituisce valore 1 (uno) se l'intersezione non risulta essere un insieme vuoto. Il risultato della funzione ST_Intersects è opposto al risultato della funzione ST_Disjoint.

La funzione ST_Intersects restituisce 1 (uno) se le condizioni di ciascuna matrice modello riportata di seguito, restituisce TRUE.

Tabella 43. Matrice per ST_Intersects (1). La funzione ST_Intersects restituisce 1 (uno) se gli spazi interni delle due forme geometriche si intersecano.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	T	*	*
Esterno forma geometrica a	*	*	*

Tabella 44. Matrice per ST_Intersects (2). La funzione ST_Intersects restituisce 1 (uno) se il contorno della prima forma geometrica si interseca con il contorno della seconda.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	*	T	*
Esterno forma geometrica a	*	*	*

Tabella 45. Matrice per ST_Intersects (3). La funzione ST_Intersects restituisce 1 (uno) se il contorno della prima forma geometrica interseca lo spazio interno della seconda.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	T	*	*
Interno forma geometrica a	*	*	*
Esterno forma geometrica a	*	*	*

Tabella 46. Matrice per ST_Intersects (4). La funzione ST_Intersects restituisce 1 (uno) se i contorni delle forme geometriche si intersecano.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	T	*
Interno forma geometrica a	*	*	*
Esterno forma geometrica a	*	*	*

ST_Crosses

ST_Crosses utilizza due forme geometriche e restituisce valore 1 (uno) se:

- L'intersezione risulta in una forma geometrica le cui dimensioni sono inferiori alla dimensione massima delle forme geometriche di origine.
- L'intersezione è interna alle due forme geometriche di origine.

ST_Crosses restituisce un valore nullo se la prima forma geometrica è una superficie o una multisuperficie o se la seconda forma geometrica è un punto o un multipunto. Per tutte le altre combinazioni, ST_Crosses restituisce valore 1 (le due forme geometriche si intersecano) o valore 0 (le forme non si intersecano).

La seguente immagine mostra multipunti che attraversano linee, linee che attraversano altre linee, multipunti che attraversano un poligono e linee che attraversano un poligono. In tre dei quattro casi, la forma geometrica *b* attraversa la forma geometrica *a*. Nel quarto caso, la forma geometrica è un multipunto che non attraversa la linea, ma tocca l'area interna al poligono *b*.

Le forme geometriche scure rappresentano la forma geometrica *a*; le forme geometriche grigie rappresentano la forma geometrica *b*.

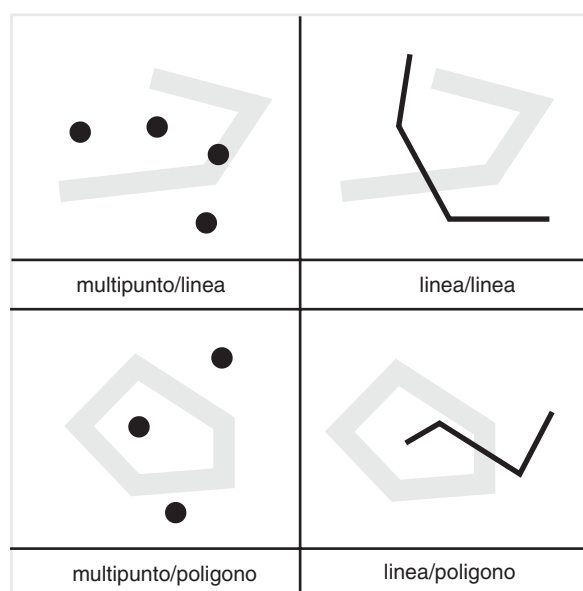


Figura 46. ST_Crosses

La matrice modello nella Tabella 47 a pagina 312 viene applicata se la prima forma geometrica è un punto o un multipunto o se è una curva o una multicurva, e la

seconda forma è una superficie. La matrice specifica che gli spazi interni devono intersecarsi e che lo spazio esterno della forma geometrica primaria (*a*) deve intersecarsi con lo spazio esterno della forma secondaria (*b*).

Tabella 47. Matrice per *ST_Crosses* (1)

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	T	*	T
Esterno forma geometrica a	*	*	*

La matrice modello nella Tabella 48 viene applicata se la prima e la seconda forma geometrica sono delle curve o multicurve. 0 indica che l'intersezione degli interni deve essere un punto (dimensione 0). Se la dimensione dell'intersezione è 1 (intersezione in una linea), la funzione *ST_Crosses* restituirà valore 0 (le forme geometriche non si intersecano); tuttavia, la funzione *ST_Overlaps* restituisce valore 1 (le forme geometriche si sovrappongono).

Tabella 48. Matrice per *ST_Crosses* (2)

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	0	*	*
Esterno forma geometrica a	*	*	*

ST_Overlaps

ST_Overlaps mette a confronto due forme geometriche con la stessa dimensione. Restituisce valore 1 (uno) se l'intersezione risulta essere una forma geometrica diversa da entrambe, ma con le stesse dimensioni.

Le forme geometriche scure rappresentano la forma geometrica *a*; le forme geometriche grigie rappresentano la forma geometrica *b*. In tutti i casi, le due forme geometriche hanno la stessa dimensione ma una si sovrappone parzialmente all'altra. L'area di sovrapposizione rappresenta la nuova forma geometrica; ha la stessa dimensione delle forme *a* e *b*.

La seguente immagine mostra sovrapposizioni di forme geometriche. I tre esempi mostrano sovrapposizioni con punti, linee e poligoni. Con i punti, i punti reali si sovrappongono. Con le linee, una parte della linea si sovrappone. Con i poligoni, una parte dell'area si sovrappone.

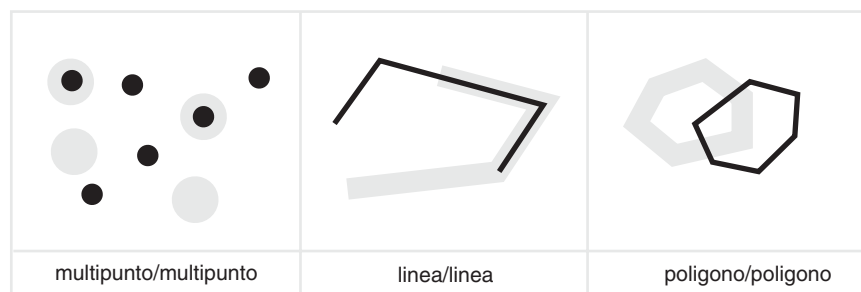


Figura 47. *ST_Overlaps*

La matrice modello nella Tabella 49 viene applicata se la prima e la seconda forma geometrica sono entrambe punti, multipunti, superfici o multisuperfici. *ST_Overlaps* restituisce valore 1 se lo spazio interno della forma geometrica si interseca con lo spazio interno ed esterno della seconda forma.

Tabella 49. *Matrice per ST_Overlaps (1)*

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	T	*	T
Esterno forma geometrica a	T	*	*

La matrice modello nella Tabella 50 viene applicata se la prima e la seconda forma geometrica sono delle curve o multicurve. In questo caso l'intersezione delle forme geometriche sarà una forma con dimensione 1 (un'altra curva). Se la dimensione dell'intersezione degli spazi interni è 0, *ST_Overlaps* restituisce valore 0 (le forme geometriche non si sovrappongono); tuttavia la funzione *ST_Crosses* restituisce valore 1 (le forme geometriche si sovrappongono).

Tabella 50. *Matrice per ST_Overlaps (2)*

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	1	*	T
Esterno forma geometrica a	T	*	*

ST_Touches

ST_Touches restituisce valore 1 (uno) se tutti i punti comuni alle due forme geometriche si trovano solo nei contorni. Gli spazi interni delle due forme geometriche non dovranno intersecarsi. Almeno una forma geometrica deve essere una curva, una superficie, una multicurve o una multisuperficie.

Le forme geometriche scure, rappresentano la forma geometrica *a*; le forme geometriche grigie rappresentano la forma *b*. In tutti i casi, il contorno della forma geometrica *b* si interseca con la forma geometrica *a*. Lo spazio interno della forma geometrica *b* è separato dalla forma geometrica *a*.

Funzioni spaziali

La seguente immagine mostra alcune esempi di contatto tra tipi di forme geometriche, ad esempio punti e linee, linee e linee, punti e poligoni, multipunti e poligoni, linee e poligoni.

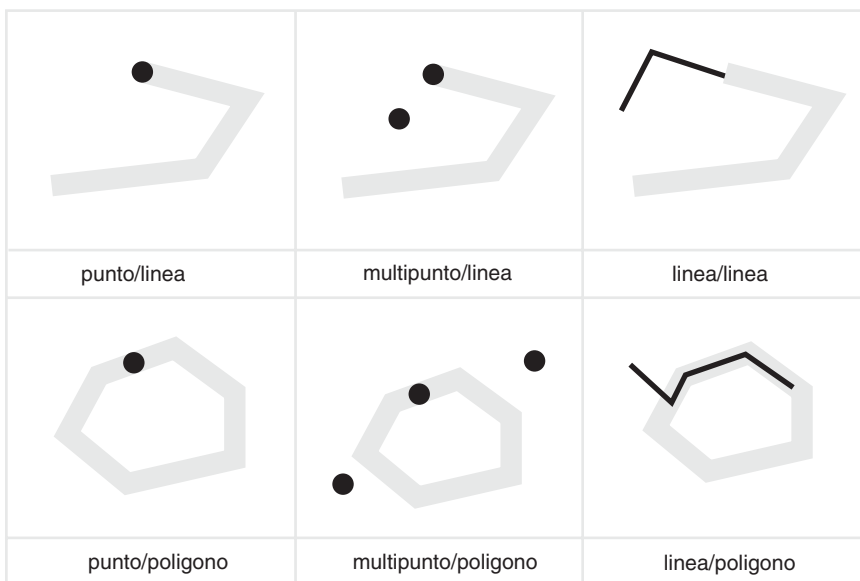


Figura 48. ST_Touches

Le matrici modello indicano che la funzione ST_Touches restituisce 1 (uno) quando gli spazi interni delle forme geometriche non si intersecano, e il contorno di ciascuna forma interseca lo spazio interno o il contorno dell'altra.

Tabella 51. Matrice per ST_Touches (1)

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	*
Interno forma geometrica a	F	T	*
Esterno forma geometrica a	*	*	*

Tabella 52. Matrice per ST_Touches (2)

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	T	*	*
Interno forma geometrica a	F	*	*
Esterno forma geometrica a	*	*	*

Tabella 53. Matrice per *ST_Touches* (3)

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	T	*
Interno forma geometrica a	F	*	*
Esterno forma geometrica a	*	*	*

Funzioni che confrontano gli involuipi delle forme geometriche

ST_EnvIntersects e *ST_MBRIntersects* sono simili in quanto determinano se il rettangolo minore che racchiude una forma geometrica si interseca con il rettangolo minore che racchiude l'altra forma. Tale rettangolo viene definito *involuppo*. I multipoligoni, i poligoni, le multilinee e le linee oblique e i rispettivi involuipi quasi coincidono; le linee orizzontali, verticali e i punti sono leggermente più piccoli dei rispettivi involuipi. *ST_EnvIntersects* stabilisce se gli involuipi delle forme geometriche si intersecano.

L'area del rettangolo minore che circonda una forma geometrica viene definito MBR (minimum bounding rectangle). Gli involuipi che circondano i multipoligoni, i poligoni le multilinee e le linee non rette, sono MBR. Gli involuipi che circondano le linee orizzontali, verticali e i punti non sono MBR, in quanto non costituiscono l'area minima a cui si adattano tali forme geometriche. Lo spazio occupato da queste ultime è definibile, quindi non avranno MBR. Tuttavia, in base a una convenzione comunemente accettata, le stesse forme geometriche rappresentano i rispettivi MBR. Quindi, rispetto ai multipoligoni, ai poligoni, alle multilinee e alle linee non rette, *ST_MBRIntersects* verifica l'intersezione degli stessi rettangoli analizzati da *ST_EnvIntersects*. Per quanto riguarda le linee orizzontali, verticali e i punti, *ST_MBRIntersects* controlla le intersezioni delle stesse forme geometriche.

ST_EnvIntersects

ST_EnvIntersects restituisce valore 1 (uno) se gli involuipi delle due forme geometriche si intersecano. Si tratta di una funzione di convenienza che implementa in modo *ST_Intersects* (*ST_Envelope*(g1), *ST_Envelope*(g2)).

ST_MBRIntersects

ST_MBRIntersects restituisce valore 1 (uno) se gli MBR (minimum bounding rectangle) delle due forme geometriche si intersecano.

Funzioni che controllano la similitudine di due elementi

ST_EqualCoordsys

ST_EqualCoordsys restituisce valore 1 (uno) se le definizioni di due sistemi di coordinate sono identiche. Durante il confronto delle definizioni, *ST_EqualCoordsys* ignora differenze quali distinzione maiuscolo minuscolo, spazi, parentesi e rappresentazione di numeri a virgola mobile.

ST_Equals

ST_Equals restituisce valore 1 (uno) se le due forme geometriche sono identiche. L'ordine dei punti utilizzati per definire le forme geometriche non è rilevante per il test di uguaglianza.

Nei sei esempi riportati (punto, multipunti, linea, multilinea, poligono e multipoligono), le forme geometriche a e b coincidono.





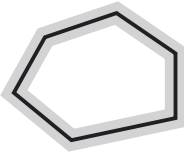
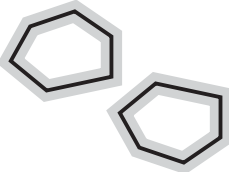
	
punto / punto	multipunto / multipunto
	
linea / linea	multilinea / multilinea
	
poligono / poligono	multipoligono / multipoligono

Figura 49. ST_Equals. Le forme geometriche scure rappresentano la forma geometrica a; le forme geometriche grigie rappresentano la forma geometrica b. In tutti i casi, la forma geometrica a è uguale alla forma geometrica b.

Tabella 54. Matrice di uguaglianza. La matrice modello DE-9IM di uguaglianza verifica che gli spazi interni si intersechino e che nessuna area interna o contorno delle forme geometriche si intersechi con lo spazio esterno delle altre.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	*	*	F
Interno forma geometrica a	T	*	F
Esterno forma geometrica a	F	F	*

ST_EqualsSRS

ST_EqualsSRS restituisce valore 1 (uno) se due sistemi di riferimento spaziali sono identici, supponendo che l'identificativo numerico di uno o di entrambi i sistemi non sia nullo.

Funzioni che controllano la presenza di intersezioni tra due forme geometriche

ST_Disjoint restituisce valore 1 (uno) se l'intersezione delle due forme geometriche è un insieme vuoto. La funzione restituisce il risultato opposto rispetto alla funzione ST_Intersects.

L'immagine mostra diverse forme geometriche e come i contorni non si intersecano in nessun punto.

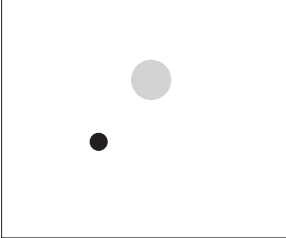
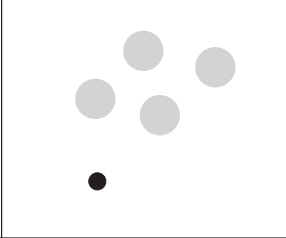
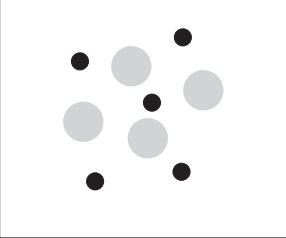
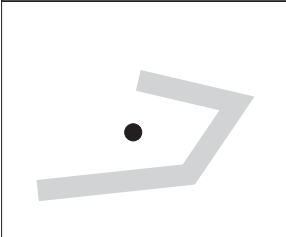

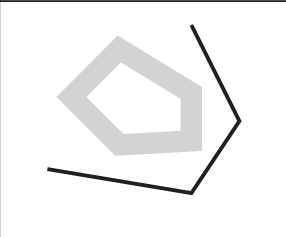
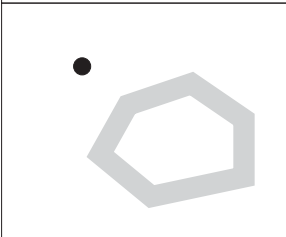
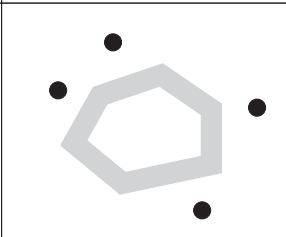
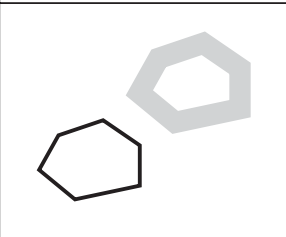
		
punto / punto	punto / multipunto	multipunto / multipunto
		
punto / linea	multilinea / linea	poligono / linea
		
punto / poligono	multipunto / multipoligono	poligono / poligono

Figura 50. ST_Disjoint. Le forme geometriche scure rappresentano la forma geometrica a; le forme geometriche grigie rappresentano la forma geometrica b. In tutti i casi, la forma geometrica a e la forma geometrica b sono staccate l'una dall'altra.

Tabella 55. Matrice ST_Disjoint. Questa matrice specifica che nessuno spazio interno o contorno delle figure geometriche si interseca.

	Interno forma geometrica b	Contorno forma geometrica b	Esterno forma geometrica b
Contorno forma geometrica a	F	F	*
Interno forma geometrica a	F	F	*
Esterno forma geometrica a	*	*	*

Funzioni che confrontano forme geometriche alla stringa matrice di modello DE-9IM

La funzione ST_Relate confronta due forme geometriche e restituisce valore 1 (uno) se le forme soddisfano le condizioni indicate dalla stringa di matrice modello DE-9IM; in caso contrario, la funzione restituisce valore 0 (zero).

Funzioni che restituiscono di informazioni sulle proprietà delle forme geometriche

Questa sezione descrive le funzioni spaziali che restituiscono le informazioni relative alle proprietà delle forme geometriche. Le informazioni comprendono:

- Tipo di dati delle forme geometriche
- Coordinate e misure di una forma geometrica
- Circonferenze, contorni, involuppi e MBR (minimum bounding rectangle)
- Dimensioni
- Se la forma è chiusa, vuota o semplice
- Forme geometriche di base appartenenti a un insieme di forme
- Sistemi di riferimento spaziali

Alcune proprietà sono a loro volta delle forme geometriche; ad esempio le circonferenze interne ed esterne di una superficie o i punti iniziali e finali di una curva. Tali forme geometriche sono prodotte da alcune delle funzioni appartenenti a questa categoria. Le funzioni che generano altri tipi di forme geometriche, ad esempio, quelle forme che rappresentano le zone che circondano una determinata area, appartengono ad un'altra categoria. Per informazioni sulla categoria "Funzioni spaziali per la generazione di nuove forme geometriche", vedere il riferimento al termine di questa sezione.

Argomenti correlati:

- "Funzioni che restituiscono informazioni sul tipo di dati" a pagina 318
- "Funzioni che restituiscono informazioni sulle coordinate e le misure" a pagina 319
- "Funzioni che restituiscono informazioni sulle forme geometriche interne a una forma geometrica" a pagina 321
- "Funzioni che mostrano informazioni su contorni, involuppi e circonferenze" a pagina 322
- "Funzioni che restituiscono informazioni sulle dimensioni di una forma geometrica" a pagina 323
- "Funzioni che rivelano se una forma geometrica è chiusa, vuota o semplice" a pagina 324
- "Funzioni che identificano il sistema di riferimento spaziale di una forma geometrica" a pagina 324

Funzioni che restituiscono informazioni sul tipo di dati

ST_GeometryType utilizza una forma geometrica come parametro di input e restituisce il nome completo del tipo dinamico della forma.

Funzioni che restituiscono informazioni sulle coordinate e le misure

Le seguenti funzioni restituiscono informazioni sulle coordinate e le misure di una forma geometrica. Ad esempio, `ST_X` può restituire la coordinata X di un determinato punto, `ST_MaxX` la coordinata X maggiore della forma geometrica e `ST_MinX` la coordinata X minore.

Queste funzioni sono:

- `ST_CoordDim`
- `ST_IsMeasured`
- `ST_IsValid`
- `ST_Is3D`
- `ST_M`
- `ST_MaxM`
- `ST_MaxX`
- `ST_MaxY`
- `ST_MaxZ`
- `ST_MinM`
- `ST_MinX`
- `ST_MinY`
- `ST_MinZ`
- `ST_X`
- `ST_Y`
- `ST_Z`

ST_CoordDim

`ST_CoordDim` restituisce un valore che indica il tipo di coordinate di una forma geometrica e se la forma contiene delle misure. Questo valore viene definito *dimensione di coordinata*. Una dimensione di coordinata differisce dalla proprietà nota come *dimensione*. Quest'ultima indica se la forma geometrica ha dimensioni come larghezza o lunghezza e non se la forma contiene coordinate relative a un tipo specifico di unità di misura.

ST_IsMeasured

`ST_IsMeasured` utilizza una forma geometrica come parametro di input e restituisce 1 se rileva nella forma le coordinate M (misure). In caso contrario, restituisce 0 (zero).

ST_IsValid

`ST_IsValid` utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è valida. In caso contrario, restituisce 0 (zero). Una forma geometrica è valida solo se tutti gli attributi nel tipo strutturato sono coerenti con la rappresentazione interna dei dati geometrici, e se la rappresentazione non è danneggiata.

ST_Is3D

`ST_Is3d` utilizza una forma geometrica come parametro di input e restituisce 1 se rileva nella forma le coordinate Z. In caso contrario, restituisce 0 (zero).

Funzioni spaziali

ST_M

Se una misura è memorizzata insieme a un punto, ST_M può utilizzare il punto come parametro di input e restituire la misura.

ST_MaxM

ST_MaxM utilizza una forma geometrica come parametro di input e ne restituisce le misure maggiori.

ST_MaxX

ST_MaxX utilizza una forma geometrica come parametro di input e restituisce le relative coordinate X maggiori.

ST_MaxY

ST_MaxY utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Y maggiori.

ST_MaxZ

ST_MaxZ utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Z maggiori.

ST_MinM

ST_MinM utilizza una forma geometrica come parametro di input e ne restituisce le misure minori.

ST_MinX

ST_MinX utilizza una forma geometrica come parametro di input e restituisce le relative coordinate X minori.

ST_MinY

ST_MinY utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Y minime.

ST_MinZ

ST_MinZ utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Z minori.

ST_X

ST_X utilizza un punto come parametro di input e restituisce la coordinata X del punto.

ST_Y

ST_Y utilizza un punto come parametro di input e restituisce la coordinata Y del punto.

ST_Z

Se una coordinata Z è memorizzata insieme a un punto, ST_Z può utilizzare il punto come parametro di input e restituire coordinata Z.

Funzioni che restituiscono informazioni sulle forme geometriche interne a una forma geometrica

Le seguenti funzioni restituiscono informazioni sulle forme geometriche interne a una forma geometrica. Alcune funzioni possono identificare determinati punti in una forma geometrica oppure restituire il numero delle forme geometriche di base appartenenti ad un insieme di forme.

Queste funzioni sono:

- ST_Centroid
- ST_EndPoint
- ST_GeometryN
- ST_LineStringN
- ST_MidPoint
- ST_NumGeometries
- ST_NumLineStrings
- ST_NumPoints
- ST_NumPolygons
- ST_PointN
- ST_PolygonN
- ST_StartPoint

ST_Centroid

ST_Centroid utilizza una forma geometrica come parametro di input e restituisce il centro geometrico, ovvero il centro del rettangolo minimo circoscritto alla forma data, come punto.

ST_EndPoint

ST_Endpoint utilizza una curva come parametro di input e restituisce l'ultimo punto della curva.

ST_GeometryN

ST_GeometryN utilizza un insieme di forme geometriche ed un indice come parametri di input e restituisce la forma geometrica nell'insieme che viene identificata dall'indice.

ST_LineStringN

ST_LineStringN utilizza una multilinea ed un indice come parametri di input e restituisce la multilinea identificata dall'indice.

ST_MidPoint

ST_MidPoint utilizza una curva come parametro di input e restituisce il punto della curva equidistante dai punti esterni.

ST_NumGeometries

ST_NumGeometries utilizza un insieme di forme geometriche come parametro di input e restituisce il numero di forme contenute nell'insieme.

ST_NumLineStrings

ST_NumLineStrings utilizza una multilinea come parametro di input e restituisce il numero delle linee che la compongono.

ST_NumPoints

ST_NumPoints utilizza una forma geometrica come parametro di input e restituisce il numero di punti utilizzati per definire la forma. Ad esempio, se la forma geometrica è un poligono definito da 5 punti, il numero restituito sarà 5.

ST_NumPolygons

ST_NumPolygons utilizza un multipoligono come parametro di input e restituisce il numero di poligoni in esso contenuti.

ST_PointN

ST_PointN utilizza una linea o un multipunto ed un indice come parametri di input e restituisce il punto della linea o del multipunto che viene identificato dall'indice.

ST_PolygonN

ST_PolygonN utilizza un multipoligono ed un indice come parametri di input e restituisce il poligono identificato dall'indice.

ST_StartPoint

ST_StartPoint utilizza una curva come parametro di input e restituisce il primo punto della curva.

Funzioni che mostrano informazioni su contorni, involucri e circonferenze

Le seguenti funzioni restituiscono informazioni sulle demarcazioni che dividono una parte interna alla forma geometrica da una parte esterna, o la forma geometrica dallo spazio esterno ad essa. Ad esempio, ST_Boundary restituisce il contorno di una forma geometrica sotto forma di curva.

Queste funzioni sono:

- ST_Boundary
- ST_Envelope
- ST_EnvIntersects
- ST_ExteriorRing
- ST_InteriorRingN
- ST_MBR
- ST_MBRIntersects
- ST_NumInteriorRing
- ST_Perimeter

ST_Boundary

ST_Boundary utilizza una forma geometrica come parametro di input e ne restituisce i contorni come nuova forma geometrica.

ST_Envelope

ST_Envelope utilizza una forma geometrica come parametro di input e restituisce l'involuppo che circonda la forma. L'involuppo è un rettangolo rappresentato come un poligono.

ST_EnvIntersects

ST_EnvIntersects utilizza due forme geometriche come parametri di input e se gli involuppi delle due forme si intersecano, restituisce 1. In caso contrario, restituisce 0 (zero).

ST_ExteriorRing

ST_ExteriorRing utilizza un poligono come parametro di input e restituisce la relativa circonferenza esterna come curva.

ST_InteriorRingN

ST_InteriorRingN utilizza un poligono ed un indice come parametri di input e restituisce la circonferenza interna identificata dall'indice dato, sotto forma di linea. Le circonferenze interne sono organizzate in base alle regole definite dalle routine di verifica delle geometrie interne.

ST_MBR

ST_MBR utilizza una forma geometrica come parametro di input e ne restituisce il rettangolo circoscritto minore.

ST_MBRIntersects

ST_MBRIntersects restituisce valore 1 (uno) se gli MBR (minimum bounding rectangle) delle due forme geometriche si intersecano.

ST_NumInteriorRing

ST_NumInteriorRing utilizza un poligono come parametro di input e restituisce il numero delle circonferenze interne al poligono.

ST_Perimeter

ST_Perimeter utilizza una superficie o una multisuperficie ed, eventualmente, un'unità di misura come parametri di input e restituisce il perimetro della superficie o multisuperficie, ovvero la lunghezza del contorno, misurato in base all'unità di misura specificata.

Funzioni che restituiscono informazioni sulle dimensioni di una forma geometrica

Le seguenti funzioni restituiscono informazioni sulle dimensioni di una forma geometrica. Ad esempio, ST_Area restituisce l'area coperta da una determinata forma geometrica.

Queste funzioni sono:

- ST_Area
- ST_Dimension
- ST_Length

ST_Area

ST_Area utilizza una forma geometrica ed eventualmente un'unità di misura come parametri di input e restituisce l'area occupata dalla forma geometrica data nell'unità di misura specificata.

ST_Dimension

ST_Dimension utilizza una forma geometrica come parametro di input e ne restituisce le dimensioni.

ST_Length

ST_Length utilizza una curva o una multicurva ed, eventualmente, un'unità di misura come parametri di input e restituisce la lunghezza della curva o multicurva data nell'unità di misura specificata.

Funzioni che rivelano se una forma geometrica è chiusa, vuota o semplice

Le seguenti funzioni indicano:

- Se una curva o una multicurva è chiusa (ovvero, se il punto iniziale e il punto finale della curva o multicurva coincidono)
- Se una forma geometrica è vuota (ovvero, sprovvista di punti)
- Se una curva, una multicurva o un punto, è semplice (ovvero, se tale forma geometrica dispone di una configurazione tipica)

ST_IsClosed

ST_IsClosed utilizza una curva o una multicurva come parametro di input e se una delle due risulta chiusa, restituisce 1. In caso contrario, restituisce 0 (zero).

ST_IsEmpty

ST_IsEmpty utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è vuota. In caso contrario, restituisce 0 (zero).

ST_IsSimple

ST_IsSimple utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è semplice. In caso contrario, restituisce 0 (zero).

Funzioni che identificano il sistema di riferimento spaziale di una forma geometrica

Le seguenti funzioni restituiscono i valori che identificano il sistema di riferimento spaziale associato alla forma geometrica. Inoltre, la funzione ST_SrsID è in grado di cambiare il sistema di riferimento spaziale della forma senza modificare o trasformare la forma geometrica.

ST_SrsId (chiamata anche ST_SRID)

ST_SrsId (o ST_SRID) utilizza una forma geometrica ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input. Gli elementi restituiti dipendono dai parametri di input specificati:

- Se viene indicato l'identificativo del sistema di riferimento spaziale, viene restituita la forma geometrica il cui sistema di riferimento spaziale è stato modificato nel sistema specificato. La forma geometrica non verrà trasformata.
- Se non viene indicato alcun identificativo di sistema di riferimento spaziale, viene restituito l'identificativo della forma geometrica data.

ST_SrsName

ST_SrsName utilizza una forma geometrica come parametro di input e restituisce il nome del sistema di riferimento spaziale in cui la forma viene rappresentata.

Funzioni che generano nuove forme geometriche da forme geometriche esistenti

Questa sezione descrive la categoria di funzioni che ottengono nuove forme geometriche da forme geometriche esistenti. La categoria non comprende le funzioni che consentono di derivare forme geometriche che rappresentano proprietà di altre forme, ma descrive le funzioni che:

- Convertono le forme geometriche in altre forme
- Creano forme geometriche che rappresentano configurazioni di spazio
- Consentono di derivare singole forme geometriche da forme multiple
- Creano forme geometriche in base a misure
- Creano modifiche di forme geometriche

Argomenti correlati:

- "Funzioni che convertono una forma geometrica in un'altra" a pagina 325
- "Funzioni che creano nuove forme geometriche con diverse configurazioni di spazio" a pagina 326
- "Funzioni che estrapolano una forma geometrica da molte" a pagina 330
- "Funzioni che estrapolano nuove forme geometriche in base alle misure" a pagina 330
- "Funzioni che creano forme modificate di forme geometriche esistenti" a pagina 331

Funzioni che convertono una forma geometrica in un'altra

Le seguenti funzioni sono in grado di convertire le forme geometriche di un super tipo nelle corrispondenti forme di tipo secondario. Ad esempio, la funzione ST_ToLineString può convertire una linea di tipo ST_Geometry in una linea di tipo ST_LineString. Alcune funzioni possono anche unire forme geometriche di base ed insiemi di forme in un singolo insieme. Ad esempio, ST_ToMultiLine è in grado di convertire una linea e una multilinea in una singola multilinea.

ST_Polygon

ST_Polygon è in grado di creare un poligono da una linea chiusa. La linea definirà la circonferenza esterna del poligono.

ST_ToGeomColl

ST_ToGeomColl utilizza una forma geometrica come parametro di input e la converte in un insieme di forme geometriche.

Funzioni spaziali

ST_ToLineString

ST_ToLineString utilizza una forma geometrica come parametro di input e la converte in una linea.

ST_ToMultiLine

ST_ToMultiLine utilizza una forma geometrica come parametro di input e la converte in una multilinea.

ST_ToMultiPoint

ST_ToMultiPoint utilizza una forma geometrica come parametro di input e la converte in un multipunto.

ST_ToMultiPolygon

ST_ToMultiPolygon utilizza una forma geometrica come parametro di input e la converte in un multipoligono.

ST_ToPoint

ST_ToPoint utilizza una forma geometrica come parametro di input e la converte in un punto.

ST_ToPolygon

ST_ToPolygon utilizza una forma geometrica come parametro di input e la converte in un poligono.

Funzioni che creano nuove forme geometriche con diverse configurazioni di spazio

Utilizzando forme geometriche esistenti come punto di partenza, le seguenti funzioni sono in grado di creare nuove forme geometriche che rappresentano aree circolari o altre configurazioni di spazio. Ad esempio, dato un punto che rappresenta il centro di un aeroporto, ST_Buffer può creare una superficie che rappresenta, in forma circolare, l'estensione prevista per l'aeroporto.

Queste funzioni sono:

- ST_Buffer
- ST_ConvexHull
- ST_Difference
- ST_Intersection
- ST_SymDifference

ST_Buffer

La funzione ST_Buffer è in grado di generare una nuova forma geometrica che parte da una forma geometrica esistente sulla base di determinato raggio. La nuova forma geometrica sarà una superficie se la forma esistente viene delimitata o se gli elementi di un insieme sono talmente vicini che le delimitazioni che circondano i singoli elementi si sovrappongono. Tuttavia, quando le aree di rispetto sono separate, ST_Buffer restituisce una multisuperficie.

La seguente figura mostra l'area di rispetto che circonda elementi singoli o sovrapposti.

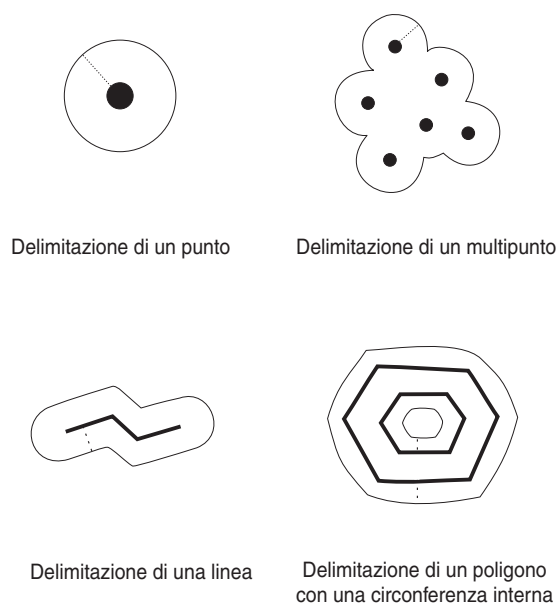


Figura 51. *ST_Buffer*

La funzione *ST_Buffer* accetta distanze positive e negative; tuttavia, solo le forme geometriche con dimensione 2 (superfici e multisuperfici) utilizzano un'area di rispetto negativa. Il valore assoluto della distanza dell'area di rispetto viene utilizzato qualora la dimensione della forma geometrica di origine è minore di 2 (tutte le forme geometriche che non siano superfici o multisuperfici).

In generale, per le circonferenze esterne, le distanze delle aree di rispetto positive generano circonferenze distanti dal centro della forma geometrica di origine; le distanze delle aree di rispetto negative generano circonferenze di superfici o multisuperfici verso il centro. Per le circonferenze interne di una superficie o multisuperficie, la distanza delle aree di rispetto positive genera circonferenze di rispetto verso il centro, mentre la distanza negativa genera un'area circolare di rispetto distaccata rispetto al centro.

Il processo di creazione delle aree di rispetto, unisce le superfici che si sovrappongono. Le distanze negative maggiori di un mezzo della larghezza massima interna di un poligono, risulteranno in forme geometriche vuote.

ST_ConvexHull

La funzione *ST_ConvexHull* restituisce il poligono convesso (convex hull) di qualsiasi forma geometrica che abbia almeno tre vertici convessi. I *Vertici* sono le coppie di coordinate X e Y contenute nelle forme geometriche. Con *poligono convesso* si intende il poligono convesso più piccolo che è possibile formare da tutti i vertici di un insieme.

La seguente figura mostra quattro esempi di poligoni convessi (convex hull). Nel primo esempio, è stata tracciata una forma irregolare simile alla lettera c. La c è chiusa dal poligono convesso. Nel quarto esempio ci sono quattro punti con linee che formano un disegno a zig-zag. La linea convessa unisce i punti quattro e due su un lato e tre e uno sull'altro.

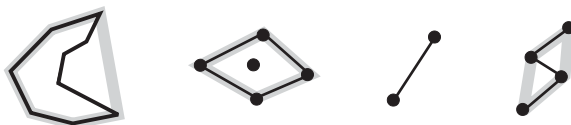


Figura 52. *ST_ConvexHull*

ST_Difference

ST_Difference utilizza due forme geometriche aventi la stessa dimensione come input. La funzione *ST_Difference* restituisce quella parte della prima forma geometrica che non si interseca con la seconda forma. Questa operazione è l'equivalente spaziale della logica AND NOT. La parte di forma geometrica restituita da *ST_Difference* è a sua volta una forma geometrica — un insieme che ha la stessa dimensione delle forme geometriche utilizzate come input. Se le due forme geometriche sono uguali, ovvero se occupano lo stesso spazio, la forma geometrica restituita sarà vuota.

A sinistra di ciascuna freccia sono raffigurate le due forme geometriche utilizzate come input da *ST_Difference*. A destra di ciascuna freccia è raffigurato l'output di *ST_Difference*. Se una parte della prima forma geometrica si interseca con la seconda, l'output sarà quella parte della prima forma geometrica non intersecata. Se le forme geometriche date sono uguali, l'output sarà una forma geometrica vuota (indicata con *nil*).

Queste figure mostrano l'input e l'output di *ST_Difference*. Ad esempio, se l'input è costituito da punti e il punto a è uguale al punto b, l'output sarà nullo. Se il punto a e il punto b sono diversi, l'output sarà un nuovo punto tra i due. Se l'input è un poligono per b ed un poligono identico ma più piccolo per la forma geometrica interna alla prima, l'output sarà nullo. Se i poligoni si sovrappongono, l'output sarà il limite più esterno dei poligoni combinati.

<p>punto / punto</p>	<p>nil</p>	<p>punto / punto</p>	<p>multipunto</p>	<p>punto / multipunto</p>	<p>multipunto</p>
<p>multipunto / multipunto</p>	<p>nil</p>	<p>multipunto / multipunto</p>	<p>multipunto</p>	<p>linea / linea</p>	<p>multilinea</p>
<p>linea / linea</p>	<p>nil</p>	<p>poligono / poligono</p>	<p>nil</p>	<p>poligono / poligono</p>	<p>poligono</p>

Figura 53. *ST_Difference*

ST_Intersection

La funzione *ST_Intersection* restituisce un insieme di punti, rappresentati come una forma geometrica, che definiscono l'intersezione di due forme geometriche date. Se le forme geometriche utilizzate da *ST_Intersection* come input non si intersecano, o

se intersecandosi, la dimensione dell'intersezione è minore alle dimensioni delle forme geometriche, *ST_Intersection* restituisce una forma geometrica vuota.

A sinistra di ciascuna freccia sono raffigurate le due forme geometriche intersecanti utilizzate come input da *ST_Intersection*. A destra di ciascuna freccia è raffigurato l'output di *ST_Intersection*, ovvero una forma geometrica che rappresenta l'intersezione creata dalle forme geometriche raffigurate a sinistra.

Questa figura illustra dieci esempi di output di *ST_Intersection*, che restituisce informazioni sulle intersezioni delle forme geometriche specificate. Ad esempio, se *b* è una linea e una forma geometrica un punto della linea, l'output sarà il multipunto in cui la forma geometrica *a* e la forma geometrica *b* convergono. Se la forma geometrica *a* e la forma geometrica *b* sono poligoni in sovrapposizione, l'output sarà un nuovo multipoligono che rappresenta solo le parti che si sovrappongono.

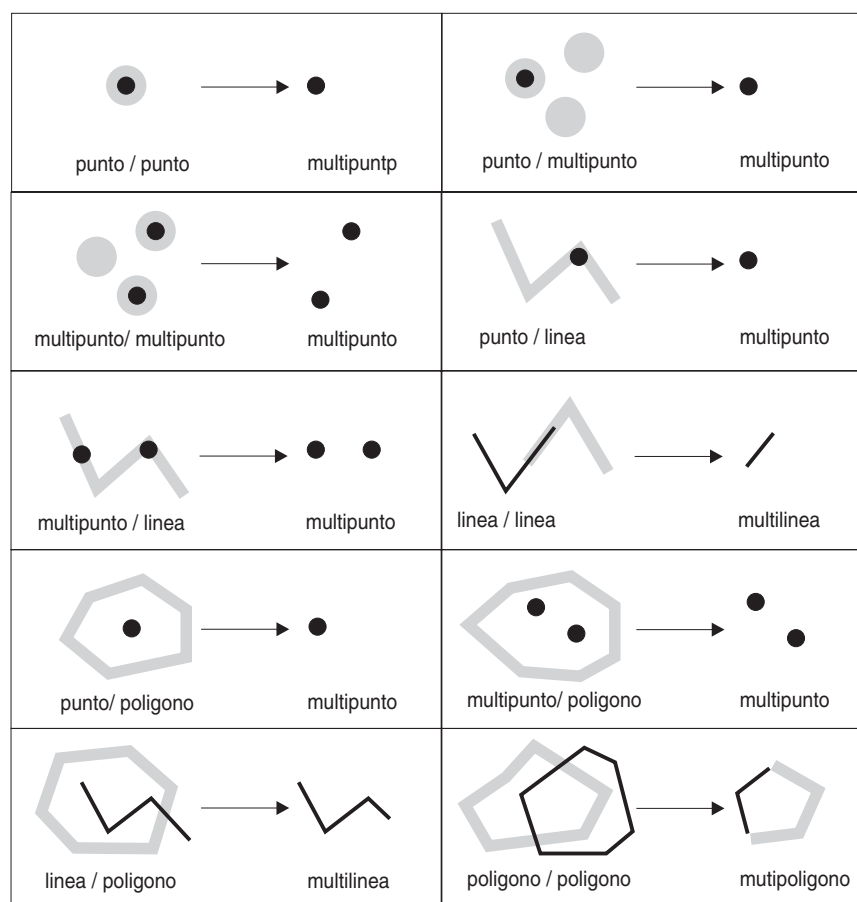


Figura 54. *ST_Intersection*

ST_SymDifference

La funzione *ST_SymDifference* restituisce la differenza simmetrica (l'equivalente spaziale dell'operazione logica XOR) di due forme geometriche intersecanti aventi stessa dimensione. Se le forme geometriche sono uguali, *ST_SymDifference* restituisce una forma geometrica vuota. In caso contrario, parte di una o di entrambe, risulterà esterna all'area dell'intersezione.

Funzioni che estrapolano una forma geometrica da molte

Le seguenti funzioni consentono di derivare singole forme geometriche da forme multiple. Ad esempio, ST_Union è in grado di unire due forme geometriche in una singola forma.

Aggregazione MBR

La combinazione delle funzioni ST_BuildMBRAggr e ST_GetAggrResult consente di aggregare una colonna di forme geometriche selezionata, in una singola forma geometrica, creando un rettangolo che rappresenta l'MBR (minimum bounding rectangle) che raccoglie tutte le forme contenute nella colonna. Le coordinate Z ed M vengono eliminate durante l'elaborazione dell'aggregazione.

ST_Union

La funzione ST_Union restituisce l'insieme dell'unione risultante da due forme geometriche. Questa operazione è l'equivalente spaziale della logica OR. Le due forme geometriche devono avere la stessa dimensione. ST_Union restituisce sempre un risultato sotto forma di insieme.

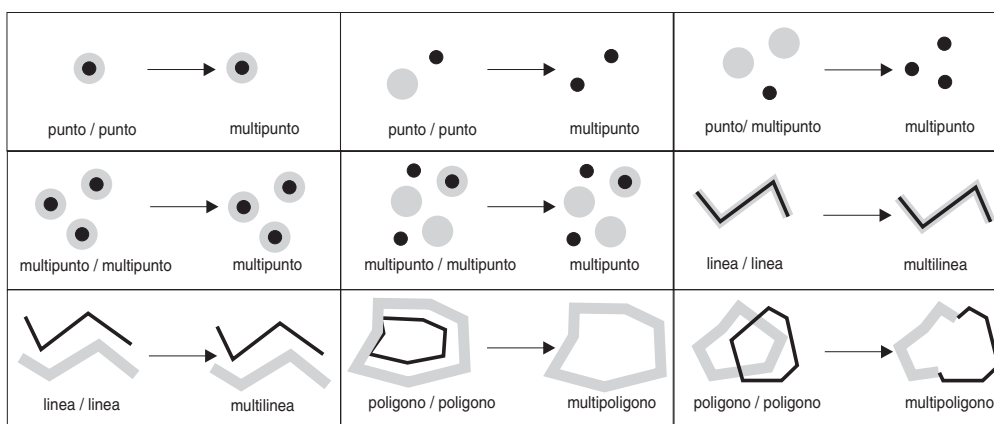


Figura 55. ST_Union

Aggregazione unione

Un'aggregazione unione è la combinazione delle funzioni ST_BuildUnionAggr e ST_GetAggrResult. Questa combinazione aggrega una colonna di forma geometriche di una tabella ad una singola forma geometrica, generando l'unione.

Funzioni che estrapolano nuove forme geometriche in base alle misure

Le funzioni illustrate in questa sezione sono in grado di creare forme geometriche i cui punti sono associati a una determinata misura o a una specifica sequenza di due misure. Ad esempio, se nei punti di una multicurva è memorizzato un intervallo di misure che varia da 4 a 8, e si desidera conoscere i punti per i quali è memorizzata la misura con valore 7, è possibile utilizzare la funzione ST_FindMeasure per restituire tali punti in un singolo multipunto.

Queste funzioni sono:

- ST_FindMeasure (chiamata anche ST_LocateAlong)
- ST_MeasureBetween (chiamata anche ST_LocateBetween)

ST_FindMeasure (chiamata anche ST_LocateAlong)

ST_FindMeasure (o ST_LocateAlong) utilizza una forma geometrica e una misura come parametri di input. Restituisce un multipunto o multicurva della forma geometrica data che corrisponde alla misura specificata. Per i punti e multipunti, vengono restituiti tutti i punti aventi la misura specificata. Per le curve, le multicurve, le superfici e le multisuperfici, per ottenere il risultato viene eseguita una interpolazione. Il calcolo delle superfici e delle multisuperfici viene eseguito in base al contorno della forma geometrica.

ST_MeasureBetween (chiamata anche ST_LocateBetween)

ST_MeasureBetween (o ST_LocateBetween) utilizzano una forma geometrica e due coordinate M (misure) come parametri di input e restituiscono la parte della forma geometrica data che rappresenta la serie di percorsi o punti disconnessi tra le due coordinate M.

Per le curve, le multicurve, le superfici e le multisuperfici, per ottenere il risultato viene eseguita una interpolazione. Nella Figura 56 i punti 3, 4, 5, 6, 7, 8 e 9 rappresentano una curva. Se le due coordinate M sono 4 e 7, ST_MeasureBetween restituirà quella parte di curva compresa tra i punti 4 e 7.

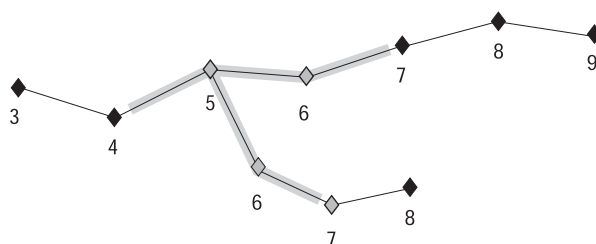


Figura 56. LocateBetween

Funzioni che creano forme modificate di forme geometriche esistenti

Le seguenti funzioni consentono di creare forme modificate di forme geometriche esistenti. Ad esempio, la funzione ST_AppendPoint crea versioni ampliate di curve esistenti. Ciascuna versione include i punti in una curva esistente più un punto aggiuntivo.

Queste funzioni sono:

- ST_AppendPoint
- ST_ChangePoint
- ST_Generalize
- ST_M
- ST_PerpPoints
- ST_RemovePoint
- ST_X
- ST_Y
- ST_Z

ST_AppendPoint

ST_AppendPoint utilizza una curva ed un punto come parametri di input ed estende la curva fino al punto indicato.

ST_ChangePoint

ST_ChangePoint utilizza una curva e due punti come parametri di input. Sostituisce tutte le ricorrenze del primo punto nella curva data con il secondo punto e restituisce la curva così ottenuta.

ST_Generalize

ST_Generalize utilizza una forma geometrica e una soglia come parametri di input e rappresenta la forma geometrica data con un numero di punti inferiore, mantenendone le caratteristiche generali. Viene utilizzato l'algoritmo di semplificazione linea Douglas-Peucker in base al quale la sequenza di punti che definisce la forma geometrica viene ripetutamente suddivisa fino a quando una serie di punti può essere sostituita da un segmento di una linea retta. Nel segmento lineare, nessuno dei punti definiti devia dalla linea retta più del limite stabilito dalla soglia data. Le coordinate Z ed M non vengono considerate nel processo di semplificazione.

ST_M

Se il punto dato non è associato ad una misura, ST_M è in grado di fornire una misura da memorizzare insieme al punto. Se al punto è associata una misura, ST_M può sostituire questa misura con una differente.

ST_PerpPoints

ST_PerpPoints utilizza una curva o una multicurva e un punto come parametri di input e restituisce la proiezione perpendicolare del punto dato sulla curva o sulla multicurva. Viene restituito il punto con la distanza inferiore tra il punto dato e il punto perpendicolare. Se due o più punti proiettati perpendicolarmente sono equidistanti dal punto dato, verranno tutti restituiti.

ST_RemovePoint

ST_RemovePoint utilizza una curva ed un punto come parametri di input e restituisce la curva data dopo aver rimosso da essa tutti i punti uguali al punto dato. Se la curva ha coordinate Z o M, il punto dovrà avere le stesse coordinate.

ST_X

ST_X è in grado di sostituire la coordinata X di un punto con un'altra coordinata X.

ST_Y

ST_Y è in grado di sostituire la coordinata Y di un punto con un'altra coordinata Y.

ST_Z

Se il punto dato non ha coordinate Z, ST_Z è in grado di aggiungere una coordinata Z al punto. Se il punto ha coordinate Z, ST_Z è in grado di sostituire questa coordinata con un'altra coordinata Z.

Funzioni che restituiscono informazioni sulla distanza

ST_Distance utilizza due forme geometriche ed eventualmente un'unità di misura come parametri di input e restituisce la distanza più breve tra i punti nella prima forma geometrica e quelli nella seconda, calcolata in base all'unità di misura specificata.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se una delle due forme geometriche date è nulla o vuota, viene restituito un valore nullo.

Ad esempio, `ST_Distance` potrebbe riportare la distanza più breve percorribile da un aereo per andare da una città all'altra. La Figura 57 contiene queste informazioni.

La figura mostra una carta geografica degli Stati Uniti con una linea retta tra Los Angeles e Chicago.



Figura 57. Distanza minima tra due città. `ST_Distance` utilizza le coordinate della posizione di Los Angeles e Chicago come input, e restituisce il valore che indica la distanza minima tra le due città.

Funzioni che restituiscono informazioni sull'indice

`ST_GetIndexParms` utilizza l'identificativo di un indice spaziale o di una colonna spaziale come parametro di input e restituisce i parametri utilizzati per definire l'indice o l'indice della colonna spaziale. Se viene specificato un ulteriore numero come parametro, viene restituito solo il parametro identificato dal numero.

Conversioni tra sistemi di coordinate

`ST_Transform` utilizza una forma geometrica e un identificativo di sistema di riferimento spaziale come parametri di input e trasforma la forma geometrica in modo che venga rappresentata nel sistema di riferimento dato. Vengono eseguite proiezioni e conversioni tra diversi sistemi di coordinate e le coordinate delle forme geometriche vengono regolate di conseguenza.

Materiale di riferimento

Capitolo 23. Funzioni spaziali: sintassi e parametri

Questa sezione introduce le funzioni spaziali descritte di seguito. Vengono descritti i fattori comuni a tutte o alla maggior parte delle funzioni spaziali. Le funzioni sono riportate in ordine alfabetico.

Funzioni spaziali: considerazioni e tipi di dati associati

In questa sezione vengono fornite informazioni necessarie per la codifica delle funzioni spaziali. Le informazioni comprendono:

- Fattori da considerare: i requisiti per specificare lo schema a cui appartengono le funzioni spaziali e il fatto che alcune funzioni possono essere richiamate come metodi.
- Come affrontare una situazione in cui una funzione spaziale non è in grado di elaborare il tipo di forme geometriche restituite da un'altra funzione spaziale.
- Una tabella che mostra quali funzioni utilizzano i valori di ciascun tipo di dati spaziali come input

Fattori da considerare

Prima di utilizzare le funzioni spaziali, tenere presente quanto segue:

- Affinché sia possibile richiamare una funzione spaziale, è necessario qualificarne il nome mediante il nome dello schema a cui le funzioni spaziali appartengono: DB2GSE. Per eseguire questa operazione, è possibile indicare esplicitamente lo schema nell'istruzione SQL a cui si fa riferimento nella funzione; ad esempio:

```
SELECT db2gse.ST_Relate (g1, g2, 'T*F***FFF2') EQUALS FROM relate_test
```

Se non si desidera specificare uno schema ogni qual volta è necessario richiamare una funzione, è possibile aggiungere DB2GSE al registro speciale CURRENT FUNCTION PATH. Per ottenere le impostazioni correnti del registro, immettere il seguente comando SQL:

```
VALUES CURRENT FUNCTION PATH
```

Per aggiornare il registro speciale CURRENT FUNCTION PATH con DB2GSE, immettere il seguente comando SQL:

```
set CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

- Alcune funzioni spaziali possono essere richiamate come metodi. Nel codice seguente, ad esempio, ST_Area viene richiamata prima come una funzione e successivamente come metodo. In entrambi in casi, la funzione ST_Area viene codificata per operare su un poligono con ID 10, memorizzato nella colonna SALES_ZONE della tabella STORES. Richiamando ST_Area, viene restituita l'area della caratteristica reale, Zona di Vendita num. 10, che il poligono rappresenta.

ST_Area richiamata come funzione:

```
SELECT ST_Area(sales_zone)
FROM   stores
WHERE  id = 10
```

ST_Area richiamata come metodo:

```
SELECT sales_zone..ST_Area()
FROM   stores
WHERE  id = 10
```

Gestione dei valori di ST_Geometry come valori di un tipo secondario

Se una funzione spaziale restituisce una forma geometrica il cui tipo statico è un super tipo, e la forma geometrica viene trasmessa ad una funzione che accetta forme di un solo tipo, secondario rispetto al super tipo, viene generata un'eccezione di durata di compilazione.

Ad esempio, il tipo statico del parametro di output della funzione ST_Union è ST_Geometry, il super tipo di tutti i tipi di dati spaziali. Il parametro di input statico per la funzione ST_PointOnSurface può essere sia ST_Polygon che ST_MultiPolygon, due tipi secondari di ST_Geometry. Se DB2® tenta di trasmettere forme geometriche restituite da ST_Union a ST_PointOnSurface, si verifica la seguente eccezione di durata di compilazione:

```
SQL00440N Nessuna funzione con nome "ST_POINTONSURFACE"  
con argomenti compatibili individuata nel percorso della funzione.  
SQLSTATE=42884
```

Il messaggio indica che DB2 non è in grado di individuare una funzione con nome ST_PointOnSurface e un parametro di input di ST_Geometry.

Affinché le forme geometriche di un super tipo possano essere trasmesse a funzioni che accettano solo tipi secondario del super tipo, utilizzare l'operatore TREAT. Come indicato in precedenza, ST_Union restituisce forme geometriche di uno dei tipi statici di ST_Geometry e forme geometriche di un tipo secondario dinamico di ST_Geometry. Se, ad esempio, restituisce una forma geometrica con un tipo dinamico di ST_MultiPolygon, l'operatore TREAT richiederebbe che questa forma geometrica venga utilizzata con il tipo statico di ST_MultiPolygon. Questo tipo corrisponde ad uno dei tipi di dati del parametro di input di ST_PointOnSurface. Se ST_Union non restituisce un valore di ST_MultiPolygon, DB2 emette un'eccezione di run-time.

Se una funzione restituisce una forma geometrica di un super tipo, l'operatore TREAT generalmente può indicare a DB2 di considerare la forma geometrica come un tipo secondario di questo super tipo. Questa operazione ha esito positivo solo se il tipo secondario corrisponde o è subordinato ad un tipo statico definito come parametro di input della funzione a cui la forma geometrica è trasmessa. Se questa condizione non è soddisfatta, DB2 emette un'eccezione di run-time.

Si consideri un'altro esempio: si desidera determinare i punti perpendicolari ad un punto dato sul contorno di un poligono continuo. Per ottenere il contorno dal poligono, viene utilizzata la funzione ST_Boundary. Il parametro di output statico di ST_Boundary è ST_Geometry, ma ST_PerpPoints accetta forme geometriche ST_Curve. Poiché il contorno di tutti i poligoni è rappresentato da una linea (che è anche una curva), e poiché il tipo di dati delle linee (ST_LineString) è secondario rispetto a ST_Curve, la seguente operazione consentirà di trasmettere un poligono ST_Geometry restituito da ST_Boundary a ST_PerpPoints:

```
SELECT ST_AsText(ST_PerpPoints(TREAT(ST_Boundary(polygon) as ST_Curve)),  
                ST_Point(30.5, 65.3, 1)))  
FROM   polygon_table
```

ST_Boundary e ST_PerpPoints invece che come funzioni, possono essere richiamate come metodi. Procedere come segue:

```
SELECT TREAT(ST_Boundary(polygon) as ST_Curve)..  
       ST_PerpPoints(ST_Point(30.5, 65.3, ))..ST_AsText()  
FROM   polygon_table
```

Funzioni spaziali elencate in base al tipo di input

La Tabella 56 contiene le funzioni spaziali elencate in base al tipo di input che possono accettare.

Importante: I tipi di dati spaziali formano una gerarchia avente ST_Geometry come elemento principale. Quando la documentazione di DB2 Spatial Extender indica che un valore di un super tipo appartenente alla gerarchia può essere utilizzato come input di una funzione, allo stesso modo anche un valore appartenente a qualsiasi tipo secondario del super tipo può essere utilizzato come input.

Ad esempio, le prime voci della Tabella 56 indicano che ST_Area e un certo numero di altre funzioni possono utilizzare i valori di del tipo di dati ST_Geometry come input. Di conseguenza, l'input di queste funzioni può anche essere costituito dai valori di qualsiasi tipo secondario di ST_Geometry: ST_Point, ST_Curve, ST_LineString e così via.

Tabella 56. Funzioni spaziali elencate in base al tipo di input

Tipo di dati del parametro di input	Funzione
ST_Geometry	EnvelopesIntersect ST_Area ST_AsBinary ST_AsGML ST_AsShape ST_AsText ST_Boundary ST_Buffer ST_BuildMBRAggr ST_BuildUnionAggr ST_Centroid ST_Contains ST_ConvexHull ST_CoordDim ST_Crosses ST_Difference ST_Dimension ST_Disjoint ST_Distance ST_Envelope ST_EnvIntersects ST_Equals ST_FindMeasure o ST_LocateAlong ST_Generalize ST_GeometryType

Considerazioni sulle funzioni spaziali

Tabella 56. Funzioni spaziali elencate in base al tipo di input (Continua)

Tipo di dati del parametro di input	Funzione
ST_Geometry, continua	ST_Intersection ST_Intersects ST_Is3D ST_IsEmpty ST_IsMeasured ST_IsSimple ST_IsValid ST_MaxM ST_MaxX ST_MaxY ST_MaxZ ST_MBR ST_MBRIntersects ST_MeasureBetween o ST_LocateBetween ST_MinM ST_MinX ST_MinY ST_MinZ ST_NumPoints ST_Overlaps ST_Relate ST_SRID o ST_SrsId ST_SrsName ST_SymDifference ST_ToGeomColl ST_ToLineString ST_ToMultiLine ST_ToMultiPoint ST_ToMultiPolygon ST_ToPoint ST_ToPolygon ST_Touches ST_Transform ST_Union ST_Within
ST_Point	ST_M ST_X ST_Y ST_Z
ST_Curve	ST_AppendPoint ST_ChangePoint ST_EndPoint ST_IsClosed ST_IsRing ST_Length ST_MidPoint ST_PerpPoints ST_RemovePoint ST_StartPoint
ST_LineString	ST_PointN ST_Polygon
ST_Surface	ST_Perimeter ST_PointOnSurface
ST_GeomCollection	ST_GeometryN ST_NumGeometries

Tabella 56. Funzioni spaziali elencate in base al tipo di input (Continua)

Tipo di dati del parametro di input	Funzione
ST_MultiPoint	ST_PointN
ST_MultiCurve	ST_IsClosed ST_Length ST_PerpPoints
ST_MultiLineString	ST_LineStringN ST_NumLineStrings ST_Polygon
ST_MultiSurface	ST_Perimeter ST_PointOnSurface
ST_MultiPolygon	ST_NumPolygons ST_PolygonN

Le funzioni ST_BuildMBRAggr e ST_BuildUnionAggr vengono descritte rispettivamente in "MBR Aggregate" e "Union Aggregate".

Riferimenti correlati:

- "Aggregazione MBR" a pagina 341
- "ST_Boundary" a pagina 352
- "ST_Area" a pagina 344
- "ST_PerpPoints" a pagina 463
- "ST_Point" a pagina 465
- "ST_PointOnSurface" a pagina 472
- "ST_Relate" a pagina 479
- "ST_Union" a pagina 497
- "Aggregazione unione" a pagina 507

EnvelopesIntersect

EnvelopesIntersect accetta due tipi di parametri di input:

- Due forme geometriche
EnvelopesIntersect restituisce 1 se l'involuppo della prima forma geometrica interseca l'involuppo della seconda forma. In caso contrario, restituisce 0 (zero).
- Una forma geometrica, quattro valori di coordinate di tipo DOUBLE che definiscono gli angoli inferiore sinistro e superiore destro di una finestra rettangolare e l'identificatore del sistema di riferimento spaziale.
EnvelopesIntersect restituisce 1 se l'involuppo della prima forma geometrica si interseca con l'involuppo definito dai quattro valori di tipo DOUBLE. In caso contrario, restituisce 0 (zero).

Sintassi:

```

▶▶ db2gse.EnvelopesIntersect (—————)
▶ —forma geometrica 1—, —forma geometrica 2— )
                        |—————|
                        | finestra-rettangolare |
    
```

Considerazioni sulle funzioni spaziali

finestra-rettangolare:

|—*x_min*—,—*y_min*—,—*x_max*—,—*y_max*—,—*srs_id*—|

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'involuppo si interseca con l'involuppo della *forma geometrica 2* o della finestra rettangolare definita dai quattro valori di tipo DOUBLE.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'involuppo si interseca con l'involuppo della *forma geometrica 1*.

x_min

Indica il valore minimo della coordinata X per l'involuppo. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

Ai dati geodetici vengono applicate le seguenti condizioni:

- *x_min* deve essere un valore di longitudine compreso tra -180 e 180 gradi.
- *x_min* è maggiore di *x_max* quando l'involuppo si sovrappone al 180° meridiano.

y_min

Indica il valore minimo della coordinata Y per l'involuppo. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

Ai dati geodetici vengono applicate le seguenti condizioni:

- *y_min* deve essere un valore di latitudine compreso tra -90 e 90 gradi.
- *y_min* deve essere inferiore al valore *y_max*.

x_max

Indica il valore massimo della coordinata X per l'involuppo. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

Ai dati geodetici vengono applicate le seguenti condizioni:

- *x_max* deve essere un valore di longitudine compreso tra -180 e 180 gradi.
- *x_max* è inferiore del valore *x_min* quando l'involuppo si sovrappone al 180° meridiano.

y_max

Indica il valore massimo della coordinata Y per l'involuppo. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è DOUBLE.

Ai dati geodetici vengono applicate le seguenti condizioni:

- *y_max* deve essere un valore di latitudine compreso tra -90 e 90 gradi.
- *y_max* deve essere maggiore al valore *y_min*.

srs_id

Identifica il sistema di riferimento spaziale. L'identificativo del sistema di

riferimento spaziale, deve corrispondere all'identificativo del sistema di riferimento spaziale del parametro della forma geometrica. Il valore del parametro non può essere nullo.

Il tipo di dati utilizzato con questo parametro è INTEGER.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare due poligoni che rappresentano delle regioni e di determinare se una di queste regioni si interseca con un'area geografica specificata dai quattro valori di tipo DOUBLE.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE counties (id INTEGER, name CHAR(20), geometry ST_Polygon)

INSERT INTO counties VALUES
(1, 'County_1', ST_Polygon('polygon((0 0, 30 0, 40 30, 40 35,
5 35, 5 10, 20 10, 20 5, 0 0))' ,0))

INSERT INTO counties VALUES
(2, 'County_2', ST_Polygon('polygon((15 15, 15 20, 60 20, 60 15,
15 15))' ,0))

INSERT INTO counties VALUES
(3, 'County_3', ST_Polygon('polygon((115 15, 115 20, 160 20, 160 15,
115 15))' ,0))

SELECT name
FROM counties as c
WHERE EnvelopesIntersect(c.geometry, 15, 15, 60, 20, 0) =1
```

Risultati:

```
Nome
-----
County_1
County_2
```

Aggregazione MBR

La combinazione delle funzioni `ST_BuildMBRAggr` e `ST_GetAggrResult` consente di aggregare una colonna di forme geometriche selezionata, in una singola forma geometrica, creando un rettangolo che rappresenta l'MBR (minimum bounding rectangle) che raccoglie tutte le forme contenute nella colonna. Le coordinate Z ed M vengono eliminate durante l'elaborazione dell'aggregazione.

Se tutte le forme geometriche da combinare hanno valore nullo, viene restituito un valore nullo. Se tutte le forme geometriche hanno valore nullo o sono vuote, viene restituita una forma geometrica vuota. Se l'MBR di tutte le forme geometriche da combinare è rappresentato da un punto, il punto viene restituito come valore `ST_Point`. Se l'MBR di tutte le forme geometriche da combinare è rappresentato da una linea orizzontale o verticale, la linea viene restituita come valore `ST_LineString`. Altrimenti, l'MBR viene restituito come valore `ST_Polygon`.

Aggregazione MBR

Sintassi:

```
► db2gse.ST_GetAggrResult (—MAX—(—  
► db2gse.ST_BuildMBRAggr (—forme geometriche—) —) —) —
```

Parametro:

forme geometriche

Una colonna selezionata che contiene un tipo di ST_Geometry o uno dei tipi secondari e rappresenta tutte le forme geometriche per cui l'MBR sarà elaborato.

Tipo restituito:

db2gse.ST_Geometry

Limitazioni:

Non è possibile creare l'aggregato di una colonna spaziale completamente selezionata nelle seguenti situazioni:

- In ambiente MPP.
- Se la clausola GROUP BY è utilizzata in modalità di selezione completa.
- Se viene utilizzata una funzione diversa dalla funzione aggregata MAX di DB2.

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio mostra come utilizzare la funzione ST_BuildMBRAggr per ottenere l'MBR di tutte le forme geometriche contenute nella colonna. In questo esempio verranno aggiunti diversi punti alla colonna GEOMETRY nella tabella SAMPLE_POINTS. Il codice SQL quindi, determinerà l'MBR di tutti i punti.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_points (id integer, geometry ST_Point)
```

```
INSERT INTO sample_points (id, geometry)  
VALUES
```

```
(1, ST_Point(2, 3, 1)),  
(2, ST_Point(4, 5, 1)),  
(3, ST_Point(13, 15, 1)),  
(4, ST_Point(12, 5, 1)),  
(5, ST_Point(23, 2, 1)),  
(6, ST_Point(11, 4, 1))
```

```
SELECT cast(ST_GetAggrResult(MAX(ST_BuildMBRAggr  
    (geometry)))..ST_AsText AS varchar(160))  
    AS ";Aggregazione_punti";  
FROM sample_points
```

Risultati:

Aggregazione_punti

```
POLYGON (( 2.00000000 2.00000000, 23.00000000 2.00000000,
23.00000000 15.00000000, 2.00000000 15.00000000, 2.00000000
2.00000000))
```

ST_AppendPoint

ST_AppendPoint utilizza una curva ed un punto come parametri di input ed estende la curva fino al punto indicato. Se la curva ha coordinate Z o M, il punto dovrà avere le stesse coordinate. La curva ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della curva data.

Se il punto da raggiungere non è rappresentato nel sistema di riferimento spaziale della curva, viene convertito nel sistema spaziale appropriato.

Se la curva data è chiusa o semplice, non è detto che la curva ottenuta come risultato sia dello stesso tipo. Se la curva e il punto hanno valore nullo, oppure se la curva è vuota, viene restituito un valore nullo. Se il punto da raggiungere è vuoto, la curva data non viene modificata e viene generato un messaggio di avviso (SQLSTATE 01HS3).

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_AppendPoint(—curva—,—punto—)—————►
```

Parametro:

curva Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la curva a cui verrà aggiunto il *punto*.

punto Un valore di tipo ST_Point che rappresenta il punto da aggiungere alla *curva*.

Tipo restituito:

db2gse.ST_Curve

Esempi:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice crea due linee, ciascuna avente tre punti.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines(id integer, line ST_Linestring)

INSERT INTO sample_lines VALUES
    (1, ST_LineString('linestring (10 10, 10 0, 0 0)', 0) )

INSERT INTO sample_lines VALUES
    (2, ST_LineString('linestring z (0 0 4, 5 5 5, 10 10 6)', 0) )
```

Esempio 1:

ST_AppendPoint

In questo esempio il punto (5, 5) viene aggiunto al termine di una linea.

```
SELECT CAST(ST_AsText(ST_AppendPoint(line, ST_Point(5, 5)))
          AS VARCHAR(120)) New
FROM   sample_lines
WHERE  id=1
```

Risultati:

NEW

```
-----
LINESTRING ( 10.00000000 10.00000000, 10.00000000 0.00000000,
0.00000000 0.00000000, 5.00000000 5.00000000)
```

Esempio 2:

In questo esempio il punto (15, 15, 7) viene aggiunto al termine di una linea con coordinate Z.

```
SELECT CAST(ST_AsText(ST_AppendPoint(line, ST_Point(15.0, 15.0, 7.0)))
          AS VARCHAR(160)) New
FROM   sample_lines
WHERE  id=2
```

Risultati:

NEW

```
-----
LINESTRING Z ( 0.00000000 0.00000000 4.00000000, 5.00000000
5.00000000 5.00000000, 10.00000000 10.00000000 6.00000000,
15.00000000 15.00000000 7.00000000)
```

ST_Area

ST_Area utilizza una forma geometrica ed eventualmente un'unità di misura come parametri di input e restituisce l'area occupata dalla forma geometrica data nell'unità di misura predefinita o specificata.

Se la forma geometrica è un poligono o un multipoligono, viene restituita l'area occupata dalla forma. L'area dei punti, delle linee, dei multipunti e delle multilinee è 0 (zero). Se la forma geometrica è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_Area (—forma geometrica— [—unità di misura—]) ►►
```

Parametri:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica che determina l'area.

unità di misura

Un valore VARCHAR(128) che rappresenta l'unità di misura in cui calcolare l'area. Le unità di misura supportate sono elencate nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Se il parametro *unità di misura* viene omissso, per determinare l'unità da utilizzare per misurare l'area, vengono applicate le seguenti regole:

- Se la *forma geometrica* si trova in un sistema di coordinate proiettate o geocentrico, verrà utilizzata l'unità di misura lineare associata al sistema.
- Se la *forma geometrica* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale, verrà utilizzata l'unità angolare associata a questo sistema di coordinate.
- Se la *forma geometrica* si trova in un SRS geodetico, l'unità di misura predefinita sarà metri quadrati.

Limitazioni sulle conversioni delle unità: se si verifica una delle seguenti condizioni, viene restituito un errore (SQLSTATE 38SU4):

- La forma geometrica si trova in un sistema di coordinate non specificato e viene specificato il parametro *unità*.
- La forma geometrica si trova in un sistema di coordinate proiettate e viene specificata un'unità angolare.
- La forma geometrica si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale geodetico, e viene specificata un'unità lineare.
- La forma geometrica si trova in un sistema di coordinate geografiche, in un SRS geodetico, e viene specificata un'unità angolare.

Tipo restituito:

DOUBLE

Esempi:

Esempio 1:

Si desidera un elenco delle aree occupate da ciascuna zona vendite. La i poligoni che rappresentano la zona vendite sono memorizzati nella tabella SAMPLE_POLYGONS. L'area viene calcolata applicando la funzione ST_Area alla colonna delle forme geometriche.

```
db2se create_srs se_bank -srsId 4000 -srsName new_york1983 -xOffset 0
      -yOffset 0 -xScale 1 -yScale 1
      -coordsysName NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

```
CREATE TABLE sample_polygons (id INTEGER, geometry ST_POLYGON)
```

```
INSERT INTO sample_polygons (id, geometry)
VALUES
  (1, ST_Polygon('polygon((0 0, 0 10, 10 10, 10 0, 0 0))', 4000) ),
  (2, ST_Polygon('polygon((20 0, 30 20, 40 0, 20 0))', 4000) ),
  (3, ST_Polygon('polygon((20 30, 25 35, 30 30, 20 30))', 4000))
```

La seguente istruzione SELECT richiama l'ID della zona vendite e l'area:

```
SELECT id, ST_Area(geometry) AS area
FROM   sample_polygons
```

Risultati:

ST_Area

ID	AREA
1	+1.00000000000000E+002
2	+2.00000000000000E+002
3	+2.50000000000000E+001

Esempio 2:

La seguente istruzione SELECT consente di richiamare l'ID della zona vendite e l'area in diverse unità di misura:

```
SELECT id,  
       ST_Area(geometry) square_feet,  
       ST_Area(geometry, 'METER') square_meters,  
       ST_Area(geometry, 'STATUTE MILE') square_miles  
FROM   sample_polygons
```

Risultati:

ID	SQUARE_FEET	SQUARE_METERS	SQUARE_MILES
1	+1.00000000000000E+002	+9.29034116132748E+000	+3.58702077598427E-006
2	+2.00000000000000E+002	+1.85806823226550E+001	+7.17404155196855E-006
3	+2.50000000000000E+001	+2.32258529033187E+000	+8.96755193996069E-007

Esempio 3:

Questo esempio consente di individuare l'area di un poligono definito nelle coordinate State Plane.

Il sistema di riferimento spaziale State con ID 3 viene creato utilizzando il seguente comando:

```
db2se create_srs SAMP_DB -srsId 3 -srsName z3101a -xOffset 0  
      -yOffset 0 -xScale 1 -yScale 1  
      -coordsysName NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

Le seguenti istruzioni SQL consentono di aggiungere il poligono, nel sistema di riferimento spaziale 3, alla tabella e di determinare l'area in piedi quadrati, metri quadrati e miglia quadrate.

```
SET current function path db2gse;  
CREATE TABLE Sample_Poly3 (id integer, geometry ST_Polygon);  
INSERT into Sample_Poly3 VALUES  
  (1, ST_Polygon('polygon((567176.0 1166411.0,  
                          567176.0 1177640.0,  
                          637948.0 1177640.0,  
                          637948.0 1166411.0,  
                          567176.0 1166411.0 ))', 3));  
SELECT id, ST_Area(geometry) "Square Feet",  
       ST_Area(geometry, 'METER') "Square Meters",  
       ST_Area(geometry, 'STATUTE MILE') "Square Miles"  
FROM Sample_Poly3;
```

Risultati:

ID	Square Feet	Square Meters	Square Miles
1	+7.94698788000000E+008	+7.38302286101346E+007	+2.85060106320552E+001

Esempio 4:

L'analista spaziale desidera un elenco dell'area coperta da ciascuna zona di esplorazione. I poligoni delle zone di esplorazione sono memorizzati nella tabella SAMPLE_GEODETIC_TAB e comprendono le seguenti zone:

- Una regione attorno al Polo Nord
- Una regione attorno al Polo Sud
- Una regione a cavallo del 180° meridiano

Il secondo campo nel seguente file di file, samp_wkt_rows.txt, contiene i poligoni che rappresentano tali zone:

```
1|'polygon((5 82,15 82,25 82,35 82,45 82,55 82,65 82,75 82,85 82,95 82,
105 82,115 82,125 82,135 82,145 82,155 82,165 82,175 82,-175 82,-165 82,
-155 82,-145 82,-135 82,-125 82,-115 82,-105 82,-95 82,-85 82,-75 82,
-65 82,-55 82,-45 82,-35 82,-25 82,-15 82,-5 82,5 82))'|'North Pole region'
2|'polygon((175 -82,165 -82,155 -82,145 -82,135 -82,125 -82,115 -82,
105 -82,95 -82,85 -82,75 -82,65 -82,55 -82,45 -82,35 -82,25 -82,15 -82,
5 -82,-5 -82,-15 -82,-25 -82,-35 -82,-45 -82,-55 -82,-65 -82,-75 -82,
-85 -82,-95 -82,-105 -82,-115 -82,-125 -82,-135 -82,-145 -82,-155 -82,
-165 -82,-175 -82,175 -82))'|'South Pole region'
3|'polygon((-175 -42,-175 1,-175 42,175 42,175 -1,175 -42,-175 -42))
'|'180th meridian'
```

Le seguenti istruzioni SQL aggiungono i poligoni, nel sistema di riferimento spaziale geodetico 2000000000, alla tabella SAMPLE_GEODETTIC_TAB.

```
SET current function path db2gse;
CREATE TABLE db2se_samp.gsege_temp_samp (
    gid          INTEGER,
    g1_wkt       varchar(500),
    comment      varchar(255)
) NOT LOGGED INITIALLY;
LOAD FROM samp_wkt_rows.txt OF DEL MODIFIED BY CHARDEL'' COLDEL|
INSERT INTO db2se_samp.gsege_temp_samp;

CREATE TABLE sample_geodetic_tab
(gid INTEGER NOT NULL PRIMARY KEY,
 geometry ST_Geometry),
comment varchar(255));

INSERT INTO sample_geodetic_tab
SELECT gid, ST_GeomFromText(g1_wkt, 2000000000), comment
FROM db2se_samp.gsege_temp_samp;
```

La funzione ST_Area calcola l'area del poligono nella colonna delle forme geometriche. L'unità di misura predefinita di ST_Area sono metri quadrati. La seguente istruzione SELECT consente di richiamare l'ID della zona di esplorazione e l'area in metri quadrati, piedi quadrati e miglia quadrate.

```
SELECT id, ST_Area(geometry) AS SQUARE_METERS,
ST_Area(geometry,'FOOT') AS SQUARE_FEET,
ST_Area(geometry,'STATUTE MILE') AS SQUARE_MILES
FROM sample_geodetic_tab
WHERE id BETWEEN 1 AND 9 ORDER BY id;
```

ID	SQUARE_METERS	SQUARE_FEET	SQUARE_MILES
1	+2.52472719957839E+012	+2.71759374028922E+013	+9.74802621488040E+005
2	+2.52475431563494E+012	+2.71762292776957E+013	+9.74813091056005E+005
3	+9.43568029137069E+012	+1.01564817377028E+014	+3.64313652781464E+006

Riferimenti correlati:

- “Funzioni spaziali supportate da DB2 Geodetic Extender” a pagina 206
- “Vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE” a pagina 293

ST_AsBinary

ST_AsBinary utilizza una forma geometrica come parametro di input e ne restituisce la rappresentazione WKB (well-known binary). Le coordinate Z ed M verranno eliminate e non saranno incluse nella rappresentazione WKB.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_AsBinary—(—forma geometrica—)—————►◄
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari, da convertire nella corrispondente rappresentazione WKB (well-known binary).

Tipo restituito:

BLOB(2G)

Esempi:

Il codice riportato di seguito indica come utilizzare la funzione ST_AsBinary per convertire i punti nella colonna delle forme geometriche della tabella SAMPLE_POINTS in rappresentazioni WKB (well-known binary) nella colonna BLOB.

```
CREATE TABLE SAMPLE_POINTS (id integer, geometry ST_POINT, wkb BLOB(32K))
```

```
INSERT INTO SAMPLE_POINTS (id, forma geometrica)
VALUES
  (1100, ST_Point(10, 20, 1))
```

Esempio 1:

L'esempio indica come inserire i dati nella colonna WKB, con ID 1111, dalla colonna GEOMETRY, con ID 1100.

```
INSERT INTO sample_points(id, wkb)
VALUES (1111,
  (SELECT ST_AsBinary(geometry)
   FROM sample_points
   WHERE id = 1100))

SELECT id, cast(ST_Point(wkb)..ST_AsText AS varchar(35)) AS point
FROM sample_points
WHERE id = 1111
```

Risultati:

```
ID punto
-----
1111 POINT ( 10.00000000 20.00000000)
```

Esempio 2:

Questo esempio consente di visualizzare la rappresentazione binaria WKB.

```
SELECT id, substr(ST_AsBinary(geometry), 1, 21) AS point_wkb
FROM   sample_points
WHERE  id = 1100
```

Risultati:

```
ID POINT_WKB
-----
1100 x'0101000000000000000000000024400000000000003440'
```

Riferimenti correlati:

- “Rappresentazione WKB (well-known binary)” a pagina 520

ST_AsGML

ST_AsGML utilizza una forma geometrica come parametro di input e ne restituisce la rappresentazione utilizzando il linguaggio GML (geography markup language).

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_AsGML—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari, da convertire nella corrispondente rappresentazione GML.

Tipo restituito:

CLOB(2G)

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il seguente frammento di codice mostra come visualizzare il frammento GML utilizzando la funzione ST_AsGML. L'esempio indica come inserire i dati nella colonna GML, dalla colonna delle forme geometriche, con ID 2222.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE SAMPLE_POINTS (id integer, geometry ST_POINT, gml CLOB(32K))

INSERT INTO SAMPLE_POINTS (id, geometry)
VALUES
  (1100, ST_Point(10, 20, 1))

INSERT INTO sample_points(id, gml)
VALUES (2222,
  (SELECT ST_AsGML(geometry)
   FROM   sample_points
   WHERE  id = 1100))
```

ST_AsGML

La seguente istruzione SELECT consente di elencare l'ID e la rappresentazione GML delle forme geometriche. La forma geometrica viene convertita in un frammento GML mediante la funzione ST_AsGML.

```
SELECT id, cast(ST_AsGML(geometry) AS varchar(110)) AS gml_fragment
FROM   sample_points
WHERE  id = 1100
```

Risultati:

L'istruzione SELECT restituisce la seguente serie di risultati:

ID	GML_FRAGMENT
1100	<gml:Point srsName";EPSG:4269";><gml:coord> <gml:X>10</gml:X><gml:Y>20</gml:Y> </gml:coord></gml:Point>

Riferimenti correlati:

- “Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati” a pagina 295
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_AsShape

ST_AsShape utilizza una forma geometrica come parametro di input e ne restituisce la rappresentazione shape ESRI.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_AsShape—(*—forma geometrica—*)—◀◀

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari, da convertire nella corrispondente rappresentazione shape ESRI.

Tipo restituito:

BLOB(2G)

Esempio:

Il frammento di codice riportato di seguito indica come utilizzare la funzione ST_AsShape per convertire i punti contenuti nella colonna delle forme geometriche della tabella SAMPLE_POINTS in rappresentazioni shape binary nella colonna shape BLOB. Mediante questo esempio i dati verranno inseriti nella colonna shape dalla colonna delle forme geometriche. La rappresentazione shape binary viene utilizzata per visualizzare le forme geometriche in geobrowser che richiedono forme geometriche compatibili con il formato ESRI, o create per il file *.SHP.


```

SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE SAMPLE_POINTS (id integer, geometry ST_POINT, shape BLOB(32K))

INSERT INTO SAMPLE_POINTS (id, geometry)
VALUES
    (1100, ST_Point(10, 20, 1))

INSERT INTO sample_points(id, shape)
VALUES (2222,
    (SELECT ST_AsShape(geometry)
     FROM sample_points
     WHERE id = 1100))

SELECT id, substr(ST_AsShape(geometry), 1, 20) AS shape
FROM sample_points
WHERE id = 1100

```

Restituisce:

```
ID      SHAPE
```

```
-----
1100 x'0100000000000000000000000024400000000000003440'
```

Riferimenti correlati:

- “Rappresentazione shape” a pagina 522

ST_AsText

ST_AsText utilizza una forma geometrica come parametro di input e ne restituisce la rappresentazione WKT (well-known text).

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_AsText—(—forma geometrica—)—————◄◄
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari, da convertire nella corrispondente rappresentazione WKT (well-known text).

Tipo restituito:

CLOB(2G)

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Dopo aver inserito i dati nella tabella SAMPLE_GEOMETRIES, per verificare che i valori inseriti siano corretti, visualizzare la rappresentazione WKT delle forme geometriche.

ST_AsText

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geometries(id SMALLINT, spatial_type varchar(18),
    geometry ST_GEOMETRY)

INSERT INTO sample_geometries(id, spatial_type, geometry)
VALUES
    (1, 'st_point', ST_Point(50, 50, 0)),
    (2, 'st_linestring', ST_LineString('linestring
    (200 100, 210 130, 220 140)', 0)),
    (3, 'st_polygon', ST_Polygon('polygon((110 120, 110 140,
    130 140, 130 120, 110 120))', 0))
```

La seguente istruzione SELECT consente di elencare il tipo spaziale e la rappresentazione WKT delle forme geometriche. La forma geometrica viene convertita in testo dalla funzione ST_AsText e trasformata in varchar(120) perché l'output della funzione ST_AsText è CLOB(2G).

```
SELECT id, tipo_spaziale, cast(geometry..ST_AsText
    AS varchar(150)) AS wkt
FROM sample_geometries
```

Risultati:

ID	SPATIAL_TYPE	WKT
1	st_point	POINT (50.00000000 50.00000000)
2	st_linestring	LINestring (200.00000000 100.00000000, 210.00000000 130.00000000, 220.00000000 140.00000000)
3	st_polygon	POLYGON ((110.00000000 120.00000000, 130.00000000 120.00000000, 130.00000000 140.00000000, 110.00000000140.00000000, 110.00000000 120.00000000))

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515

ST_Boundary

ST_Boundary utilizza una forma geometrica come parametro di input e ne restituisce i contorni come nuova forma geometrica. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se la forma geometrica data è costituita da un punto, da multipunti o da una multicurva chiusa, o se la forma è vuota, come risultato si otterrà una forma geometrica vuota di tipo ST_Point. Per le curve o le multicurve non chiuse, i punti iniziali e finali delle curve vengono restituiti con valore ST_MultiPoint a meno che tali punti non siano il punto iniziale o finale di un numero n di curve. Per le superfici o le multisuperfici, viene restituita la curva che definisce i contorni della forma geometrica data con valore ST_Curve o ST_MultiCurve. Se la forma geometrica data è nulla, viene restituito un valore nullo.

Se possibile, il tipo specifico della forma geometrica restituita sarà ST_Point, ST_LineString o ST_Polygon. Ad esempio, i contorni di un poligono senza interruzioni, sarà una linea singola rappresentata come ST_LineString. I contorni di un poligono con più interruzioni sono costituiti da più linee, rappresentate come ST_MultiLineString.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_Boundary—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari. Vengono restituiti i contorni di tale forma geometrica.

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio consente di creare diverse forme geometriche e di determinare il contorno di ciascuna forma.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Polygon('polygon((40 120, 90 120, 90 150, 40 150, 40 120))', 0))

INSERT INTO sample_geoms VALUES
  (2, ST_Polygon('polygon((40 120, 90 120, 90 150, 40 150, 40 120),
    (70 130, 80 130, 80 140, 70 140, 70 130))', 0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('linestring(60 60, 65 60, 65 70, 70 70)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('multilinestring((60 60, 65 60, 65 70, 70 70),
    (80 80, 85 80, 85 90, 90 90),
    (50 50, 55 50, 55 60, 60 60))' ,0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('point(30 30)' ,0))

SELECT id, CAST(ST_AsText(ST_Boundary(geometry)) as VARCHAR(320)) Boundary
FROM   sample_geoms
```

Risultati

ID	BOUNDARY
1	LINESTRING (40.00000000 120.00000000, 90.00000000 120.00000000, 90.00000000 150.00000000, 40.00000000 150.00000000, 40.00000000 120.00000000)
2	MULTILINESTRING ((40.00000000 120.00000000, 90.00000000 120.00000000, 90.00000000 150.00000000, 40.00000000 150.00000000, 40.00000000 120.00000000), (70.00000000 130.00000000, 80.00000000 130.00000000, 80.00000000 140.00000000, 70.00000000 140.00000000, 70.00000000 130.00000000))
3	MULTIPOINT (60.00000000 60.00000000, 65.00000000 60.00000000, 65.00000000 70.00000000, 70.00000000 70.00000000)
4	MULTIPOINT (50.00000000 50.00000000, 55.00000000 50.00000000, 55.00000000 60.00000000, 60.00000000 60.00000000, 60.00000000 50.00000000, 65.00000000 50.00000000, 65.00000000 60.00000000, 70.00000000 60.00000000, 70.00000000 70.00000000, 80.00000000 80.00000000, 85.00000000 80.00000000, 85.00000000 90.00000000, 90.00000000 90.00000000)
5	POINT EMPTY

ST_Buffer

ST_Buffer utilizza una forma geometrica, una distanza ed eventualmente un'unità di misura come parametri di input e restituisce la forma geometrica che circonda la forma geometrica data in base alla distanza specificata, calcolata nell'unità di misura indicata. Ciascun punto del contorno della forma geometrica ottenuta come risultato, è la distanza specificata dalla geometria data. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Per i dati geodetici, se viene specificata una distanza negativa, ST_Buffer restituisce una regione più lontana da tutti i punti nella forma geometrica di input, rispetto alla distanza specificata. In altre parole, una distanza negativa, restituisce una regione complementare.

Eventuali curve circolari presenti nel contorno della forma geometrica ottenuta come risultato viene approssimata con linee. Ad esempio, l'area di rispetto che circonda un punto, costituita da un'area circolare, viene approssimata utilizzando un poligono il cui contorno è formato da linee.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```

▶▶—db2gse.ST_Buffer—————▶
▶—(—forma geometrica—,—distanza—, —unità di misura—)————▶

```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica intorno alla quale creare l'area di rispetto. Per i dati geodetici, ST_Buffer supporta solo i tipi di dati ST_Point e ST_MultiPoint.

distanza

Un valore DOUBLE PRECISION che indica la distanza da utilizzare per creare l'area di rispetto della *forma geometrica*. Per i dati geodetici, la distanza non deve essere superiore al raggio equatoriale della Terra. Per l'ellissoide WGS-84, questa lunghezza equivale a 6378137,0 metri.

unità di misura

Un valore VARCHAR(128) che identifica l'unità di misura in cui calcolare la *distanza*. Le unità di misura supportate sono elencate nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Se il parametro *unità di misura* viene omesso, per determinare l'unità da utilizzare per la distanza vengono applicate le seguenti regole:

- Se la *una forma geometrica* si trova in un sistema di coordinate proiettate o geocentrico, l'unità di misura lineare associata al sistema sarà quella predefinita.
- Se la *forma geometrica* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale, l'unità angolare associata a questo sistema di coordinate sarà utilizzata come predefinita.

- Se la *forma geometrica* si trova in un SRS geodetico, l'unità di misura predefinita sarà metri.

Limitazioni sulle conversioni delle unità: se si verifica una delle seguenti condizioni, viene restituito un errore (SQLSTATE 38SU4):

- La forma geometrica si trova in un sistema di coordinate non specificato e viene specificato il parametro *unità*.
- La forma geometrica si trova in un sistema di coordinate proiettate e viene specificata un'unità angolare.
- La forma geometrica si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale geodetico, e viene specificata un'unità lineare.
- La forma geometrica si trova in un sistema di coordinate geografiche, in un SRS geodetico, e viene specificata un'unità angolare.

Tipo restituito:

db2gse.ST_Geometry

Esempi:

Nei seguenti esempi, i risultati sono stati formattati nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito consente di creare un sistema di riferimento spaziale, la tabella SAMPLE_GEOMETRIES e di inserire i dati nella tabella.

```
db2se create_srs se_bank -srsId 4000 -srsName new_york1983
-xOffset 0 -yOffset 0 -xScale 1 -yScale 1
-coordsysName NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE
  sample_geometries (id INTEGER, spatial_type varchar(18),
  geometry ST_GEOMETRY)
```

```
INSERT INTO sample_geometries(id, spatial_type, geometry)
VALUES
  (1, 'st_point', ST_Point(50, 50, 4000)),
  (2, 'st_linestring',
  ST_LineString('linestring(200 100, 210 130,
  220 140)', 4000)),
  (3, 'st_polygon',
  ST_Polygon('polygon((110 120, 110 140, 130 140,
  130 120, 110 120))',4000)),
  (4, 'st_multipolygon',
  ST_MultiPolygon('multipolygon(((30 30, 30 40,
  35 40, 35 30, 30 30),(35 30, 35 40, 45 40,
  45 30, 35 30)))', 4000))
```

Esempio 1:

La seguente istruzione SELECT utilizza la funzione ST_Buffer per applicare un'area di rispetto pari a 10.

```
SELECT id, spatial_type,
  cast(geometry..ST_Buffer(10)..ST_AsText AS varchar(470)) AS buffer_10
FROM   sample_geometries
```

Risultati:

ST_Buffer

ID	SPATIAL_TYPE	BUFFER_10
1	st_point	POLYGON ((60.00000000 50.00000000, 59.00000000 55.00000000, 54.00000000 59.00000000, 49.00000000 60.00000000, 44.00000000 58.00000000, 41.00000000 53.00000000, 40.00000000 48.00000000, 42.00000000 43.00000000, 47.00000000 41.00000000, 52.00000000 40.00000000, 57.00000000 42.00000000, 60.00000000 50.00000000))
2	st_linestring	POLYGON ((230.00000000 140.00000000, 229.00000000 145.00000000, 219.00000000 150.00000000, 213.00000000 147.00000000, 203.00000000 137.00000000, 201.00000000 133.00000000, 191.00000000 103.00000000, 191.00000000 99.00000000, 192.00000000 95.00000000, 196.00000000 91.00000000, 200.00000000 91.00000000, 204.00000000 91.00000000, 209.00000000 97.00000000, 218.00000000 124.00000000, 227.00000000 133.00000000, 230.00000000 140.00000000))
3	st_polygon	POLYGON ((140.00000000 120.00000000, 140.00000000 140.00000000, 139.00000000 145.00000000, 130.00000000 150.00000000, 110.00000000 150.00000000, 105.00000000 149.00000000, 100.00000000 140.00000000, 100.00000000 120.00000000, 101.00000000 115.00000000, 110.00000000 110.00000000, 130.00000000 110.00000000, 135.00000000 111.00000000, 140.00000000 120.00000000))
4	st_multipolygon	POLYGON ((55.00000000 30.00000000, 55.00000000 40.00000000, 54.00000000 45.00000000, 45.00000000 50.00000000, 30.00000000 50.00000000, 25.00000000 49.00000000, 20.00000000 40.00000000, 20.00000000 30.00000000, 21.00000000 25.00000000, 30.00000000 20.00000000, 45.00000000 20.00000000, 50.00000000 21.00000000, 55.00000000 30.00000000))

Esempio 2:

La seguente istruzione SELECT utilizza la funzione ST_Buffer per applicare un'area di rispetto negativa pari a 5.

```
SELECT id, spatial_type,
       cast(ST_AsText(ST_Buffer(geometry, -5)) AS varchar(150))
       AS buffer_negative_5
FROM   sample_geometries
WHERE  id = 3
```

Risultati:

ID	SPATIAL_TYPE	BUFFER_NEGATIVE_5
3	st_polygon	POLYGON ((115.00000000 125.00000000, 125.00000000 125.00000000, 125.00000000 135.00000000, 115.00000000 135.00000000, 115.00000000 125.00000000))

Esempio 3:

La seguente istruzione SELECT mostra il risultato dell'applicazione di un'area di rispetto avendo specificato il parametro unità di misura.

```
SELECT id, spatial_type,
       cast(ST_AsText(ST_Buffer(geometry, 10, 'METER')) AS varchar(680))
       AS buffer_10_meter
FROM   sample_geometries
WHERE  id = 3
```

Risultati:

ID	SPATIAL_TYPE	BUFFER_10_METER
3	st_polygon	POLYGON ((163.00000000 120.00000000, 163.00000000 140.00000000, 162.00000000 149.00000000, 159.00000000 157.00000000, 152.00000000 165.00000000, 143.00000000 170.00000000, 130.00000000 173.00000000, 110.00000000 173.00000000, 101.00000000 172.00000000, 92.00000000 167.00000000, 84.00000000 160.00000000, 79.00000000 151.00000000, 77.00000000 140.00000000, 77.00000000 120.00000000, 78.00000000 111.00000000, 83.00000000 102.00000000, 90.00000000 94.00000000, 99.00000000 89.00000000, 110.00000000 87.00000000, 130.00000000 87.00000000, 139.00000000 88.00000000, 147.00000000 91.00000000, 155.00000000 98.00000000, 160.00000000 107.00000000, 163.00000000 120.00000000))

Riferimenti correlati:

- “Funzioni spaziali supportate da DB2 Geodetic Extender” a pagina 206
- “Vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE” a pagina 293

ST_Centroid

ST_Centroid utilizza una forma geometrica come parametro di input e restituisce il centro geometrico, ovvero il centro del rettangolo minimo circoscritto alla forma data, come punto. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_Centroid—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica per determinare il centro geometrico.

Tipo restituito:

db2gse.ST_Point

Esempio:

L'esempio consente di creare due forme geometriche e ne individua il centroide.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms VALUES
(1, ST_Polygon('polygon
((40 120, 90 120, 90 150, 40 150, 40 120),
(50 130, 80 130, 80 140, 50 140, 50 130))',0))
```

```
INSERT INTO sample_geoms VALUES
(2, ST_MultiPoint('multipoint(10 10, 50 10, 10 30)',0))
```

ST_Centroid

```
SELECT id, CAST(ST_AsText(ST_Centroid(geometry))
as VARCHAR(40)) Centroid
FROM sample_geoms
```

Risultati:

ID	CENTROID
1	POINT (65.00000000 135.00000000)
2	POINT (30.00000000 20.00000000)

ST_ChangePoint

ST_ChangePoint utilizza una curva e due punti come parametri di input. Sostituisce tutte le ricorrenze del primo punto nella curva data con il secondo punto e restituisce la curva così ottenuta. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se i due punti non sono rappresentati nel sistema di riferimento spaziale della curva, verranno convertiti nel sistema spaziale appropriato.

Se la curva è vuota, viene restituito un valore nullo. Se la curva o uno dei punti dati hanno valore nullo o sono vuoti, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_ChangePoint(—curva—, —vecchio_punto—, —nuovo_punto—)
```

Parametro:

curva Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la curva in cui i punti identificati con *vecchio_punto* vengono modificati in *nuovo_punto*.

vecchio_punto

Un valore di tipo ST_Point che identifica i punti della curva che vengono modificati in *nuovo_punto*.

nuovo_punto

Un valore di tipo ST_Point che rappresenta la nuova posizione dei punti della curva identificati da *vecchio_punto*.

Tipo restituito:

db2gse.ST_Curve

Limitazioni:

Il punto da modificare nella curva deve essere uno dei punti utilizzati per definire la curva.

Se la curva ha coordinate Z o M, i punti dati dovranno avere le stesse coordinate.

Esempi:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito consente di creare la tabella SAMPLE_LINES e di inserirvi i dati.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines(id INTEGER, line ST_LineString)

INSERT INTO sample_lines VALUES
  (1, ST_LineString('linestring (10 10, 5 5, 0 0, 10 0, 5 5, 0 10)', 0) )

INSERT INTO sample_lines VALUES
  (2, ST_LineString('linestring z (0 0 4, 5 5 5, 10 10 6, 5 5 7)', 0) )
```

Esempio 1:

Questo esempio consente di modificare tutte le ricorrenze del punto (5, 5) nel punto (6, 6) della linea.

```
SELECT cast(ST_AsText(ST_ChangePoint(line, ST_Point(5, 5),
                                     ST_Point(6, 6))) as VARCHAR(160))
FROM   sample_lines
WHERE  id=1
```

Risultati:

```
NEW
-----
LINESTRING ( 10.00000000 10.00000000, 6.00000000 6.00000000, 0.00000000
0.00000000, 10.00000000 0.00000000, 6.00000000 6.00000000, 0.00000000
10.00000000)
```

Esempio 2:

Questo esempio consente di modificare tutte le ricorrenze del punto (5, 5, 5) nel punto (6, 6, 6) della linea.

```
SELECT cast(ST_AsText(ST_ChangePoint(line, ST_Point(5.0, 5.0, 5.0),
                                     ST_Point(6.0, 6.0, 6.0) )) as VARCHAR(180))
FROM   sample_lines
WHERE  id=2
```

Risultati:

```
NEW
-----
LINESTRING Z ( 0.00000000 0.00000000 4.00000000, 6.00000000 6.00000000
6.00000000, 10.00000000 10.00000000 6.00000000, 5.00000000 5.00000000
7.00000000)
```

ST_Contains

ST_Contains utilizza due forme geometriche come parametri di input e restituisce 1 se la prima forma geometrica contiene completamente la seconda; in caso contrario, per indicare che la prima forma non contiene completamente la seconda, restituisce 0 (zero).

Se una delle forme geometriche date è nulla o vuota, viene restituito un valore nullo.

ST_Contains

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

Sintassi:

►►—db2gse.ST_Contains—(—*forma geometrica 1*—,—*forma geometrica 2*—)————►►

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per verificare se contiene completamente la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per verificare se è contenuta completamente nella *forma geometrica 1*.

Limitazioni: per i dati geodetici, entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Tipo restituito:

INTEGER

Esempi:

Il codice riportato di seguito consente di creare le tabelle e di inserirvi i dati.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_points(id SMALLINT, geometry ST_POINT)

CREATE TABLE sample_lines(id SMALLINT, geometry ST_LINESTRING)

CREATE TABLE sample_polygons(id SMALLINT, geometry ST_POLYGON)

INSERT INTO sample_points (id, geometry)
VALUES
  (1, ST_Point(10, 20, 1)),
  (2, ST_Point('point(41 41)', 1))

INSERT INTO sample_lines (id, geometry)
VALUES
  (10, ST_LineString('linestring (1 10, 3 12, 10 10)', 1) ),
  (20, ST_LineString('linestring (50 10, 50 12, 45 10)', 1) )
INSERT INTO sample_polygons(id, geometry)
VALUES
  (100, ST_Polygon('polygon((0 0, 0 40, 40 40, 40 0, 0 0))', 1) )
```

Esempio 1:

Il codice riportato di seguito utilizza la funzione ST_Contains per individuare i punti contenuti in un determinato poligono.

```
SELECT poly.id AS id_poligono,
       CASE ST_Contains(poly.geometry, pts.geometry)
         WHEN 0 THEN 'does not contain'
         WHEN 1 THEN 'does contain'
```

```

        END AS contains,
        pts.id AS id_punto
FROM    sample_points pts, sample_polygons poly

```

Risultati:

POLYGON_ID CONTAINS	POINT_ID
100 does contain	1
100 does not contain	2

Esempio 2:

Il codice riportato di seguito utilizza la funzione ST_Contains per individuare le linee contenute in un determinato poligono.

```

SELECT poly.id AS id_poligono,
       CASE ST_Contains(poly.geometry, line.geometry)
         WHEN 0 THEN 'does not contain'
         WHEN 1 THEN 'does contain'
       END AS contains,
       line.id AS id_linea
FROM    sample_lines line, sample_polygons poly

```

Risultati:

POLYGON_ID CONTAINS	LINE_ID
100 does contain	10
100 does not contain	20

Riferimenti correlati:

- “ST_Within” a pagina 499

ST_ConvexHull

ST_ConvexHull utilizza una forma geometrica come parametro di input e ne restituisce il poligono convesso (convex hull).

La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se possibile, il tipo specifico della forma geometrica restituita sarà ST_Point, ST_LineString o ST_Polygon. Ad esempio, i contorni di un poligono senza interruzioni, sarà una linea singola rappresentata come ST_LineString. I contorni di un poligono con più interruzioni sono costituiti da più linee, rappresentate come ST_MultiLineString.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```

▶▶—db2gse.ST_ConvexHull—(—forma geometrica—)————▶▶

```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da cui calcolare il poligono convesso (convex hull).

ST_ConvexHull

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito consente di creare la tabella SAMPLE_GEOMETRIES e di inserirvi i dati.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geometries(id INTEGER, spatial_type varchar(18),  
    geometry ST_GEOMETRY)
```

```
INSERT INTO sample_geometries(id, spatial_type, geometry)  
VALUES  
  (1, 'ST_LineString', ST_LineString  
    ('linestring(20 20, 30 30, 20 40, 30 50)', 0)),  
  (2, 'ST_Polygon', ST_Polygon('polygon  
    ((110 120, 110 140, 120 130, 110 120))', 0) ),  
  (3, 'ST_Polygon', ST_Polygon('polygon((30 30, 25 35, 15 50,  
    35 80, 40 85, 80 90,70 75, 65 70, 55 50, 75 40, 60 30,  
    30 30))', 0) ),  
  (4, 'ST_MultiPoint', ST_MultiPoint('multipoint(20 20, 30 30,  
    20 40, 30 50)', 1))
```

La seguente istruzione SELECT calcola il poligono convesso (convex hull) di tutte le forme geometriche create precedentemente e visualizza il risultato.

```
SELECT id, spatial_type, cast(geometry..ST_ConvexHull..ST_AsText  
    AS varchar(300)) AS convexhull  
FROM sample_geometries
```

Risultati:

ID	SPATIAL_TYPE	CONVEXHULL
1	ST_LineString	POLYGON ((20.00000000 40.00000000, 20.00000000 20.00000000, 30.00000000 30.00000000, 30.00000000 50.00000000, 20.00000000 40.00000000))
2	ST_Polygon	POLYGON ((110.00000000 140.00000000, 110.00000000 120.00000000, 120.00000000 130.00000000, 110.00000000 140.00000000))
3	ST_Polygon	POLYGON ((15.00000000 50.00000000, 25.00000000 35.00000000, 30.00000000 30.00000000, 60.00000000 30.00000000, 75.00000000 40.00000000, 80.00000000 90.00000000, 40.00000000 85.00000000, 35.00000000 80.00000000, 15.00000000 50.00000000))
4	ST_MultiPoint	POLYGON ((20.00000000 40.00000000, 20.00000000 20.00000000, 30.00000000 30.00000000, 30.00000000 50.00000000, 20.00000000 40.00000000))

ST_CoordDim

ST_CoordDim utilizza una forma geometrica come parametro di input e restituisce la dimensionalità delle coordinate.

Se la forma geometrica data non ha coordinate Z e M, la dimensionalità sarà 2. Se ha coordinate Z e nessuna coordinata M o viceversa, la dimensionalità sarà 3. Se ha coordinate Z ed M, la dimensionalità sarà 4. Se la forma geometrica ha valore nullo, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_CoordDim—(—*forma geometrica*—)—————►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da cui ricavare la dimensionalità.

Tipo restituito:

INTEGER

Esempio:

L'esempio consente di creare diverse forme geometriche e di determinare la dimensionalità delle coordinate.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id CHARACTER(15), geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  ('Empty Point', ST_Geometry('point EMPTY',0))

INSERT INTO sample_geoms VALUES
  ('Linestring', ST_Geometry('linestring (10 10, 15 20)',0))

INSERT INTO sample_geoms VALUES
  ('Polygon', ST_Geometry('polygon((40 120, 90 120, 90 150,
  40 150, 40 120))' ,0))

INSERT INTO sample_geoms VALUES
  ('Multipoint M', ST_Geometry('multipoint m (10 10 5, 50 10
  6, 10 30 8)' ,0))

INSERT INTO sample_geoms VALUES
  ('Multipoint Z', ST_Geometry('multipoint z (47 34 295,
  23 45 678)' ,0))

INSERT INTO sample_geoms VALUES
  ('Point ZM', ST_Geometry('point zm (10 10 16 30)' ,0))

SELECT id, ST_CoordDim(geometry) COORDDIM
FROM sample_geoms
```

Risultati:

ST_CoordDim

ID	COORDDIM
Empty Point	2
Linestring	2
Polygon	2
Multipoint M	3
Multipoint Z	3
Point ZM	4

ST_Crosses

ST_Crosses utilizza due forme geometriche come parametri di input e se le due forme si intersecano, restituisce 1. In caso contrario, restituisce 0 (zero).

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se la prima forma è un poligono o un multipoligono, o se la seconda forma è un punto o un multipunto, o se una delle forme ha valore nullo o è vuota, viene restituito un valore nullo. Se l'intersezione delle due forme geometriche è una forma geometrica la cui dimensione ha un valore in meno rispetto alla dimensione massima delle due forme date, e se la forma risultante non è uguale a nessuna delle due forme date, viene restituito 1. In caso contrario, viene restituito 0 (zero).

Sintassi:

```
db2gse.ST_Crosses(—forma geometrica 1—,—forma geometrica 2—)
```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina l'intersezione con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per verificarne l'intersezione con la *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempio:

Il codice riportato di seguito consente di determinare se le forme geometriche create si intersecano l'una con l'altra.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('polygon((30 30, 30 50, 50 50, 50 30, 30 30))', 0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('linestring(40 50, 50 40)', 0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('linestring(20 20, 60 60)', 0))
```

```
SELECT a.id, b.id, ST_Crosses(a.geometry, b.geometry) Crosses
FROM   sample_geoms a, sample_geoms b
```

Risultati:

ID	ID	CROSSES
1	1	-
2	1	0
3	1	1
1	2	-
2	2	0
3	2	1
1	3	-
2	3	1
3	3	0

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_Difference

ST_Difference utilizza due forme geometriche come parametri di input e restituisce la parte della prima forma geometrica che non si interseca con la seconda forma geometrica.

Entrambe le forme geometriche devono essere della stessa dimensione. Se la forma geometrica è nulla, viene restituito un valore nullo. Se la prima forma geometrica è vuota, viene restituita una forma geometrica vuota di tipo ST_Point. Se la seconda forma geometrica è vuota, la prima forma geometrica viene restituita non modificata.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_Difference(—forma geometrica 1—, —forma geometrica 2—) ◀◀
```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry che rappresenta la prima forma geometrica da utilizzare per calcolare la differenza con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry che rappresenta la seconda forma geometrica da utilizzare per calcolare la differenza con la *forma geometrica 1*.

Limitazioni per i dati geodetici:

ST_Difference

- Entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
- ST_Difference supporta solo i tipi di dati ST_Point, ST_Polygon, ST_MultiPoint e ST_MultiPolygon.

Tipo restituito:

db2gse.ST_Geometry

La dimensione della forma geometrica restituita è uguale alla dimensione delle forme geometriche di input.

Esempi:

Nel seguente esempio, i risultati sono stati formattati nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito consente di creare la tabella SAMPLE_GEOMETRIES e di inserirvi i dati.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('polygon((10 10, 10 20, 20 20, 20 10, 10 10))' ,0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((30 30, 30 50, 50 50, 50 30, 30 30))' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('polygon((40 40, 40 60, 60 60, 60 40, 40 40))' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring(70 70, 80 80)' ,0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('linestring(75 75, 90 90)' ,0))
```

Esempio 1:

Questo esempio consente di individuare la differenza tra due poligoni disgiunti.

```
SELECT a.id, b.id, CAST(ST_AsText(ST_Difference(a.geometry, b.geometry))
  as VARCHAR(200)) Difference
FROM sample_geoms a, sample_geoms b
WHERE a.id = 1 and b.id = 2
```

Risultati:

ID	ID	DIFFERENCE
1	2	POLYGON ((10.00000000 10.00000000, 20.00000000 10.00000000, 20.00000000 20.00000000, 10.00000000 20.00000000, 10.00000000 10.00000000))

Esempio 2:

Questo esempio consente di individuare la differenza tra due poligoni intersecanti.

```
SELECT a.id, b.id, CAST(ST_AsText(ST_Difference(a.geometry, b.geometry))
  as VARCHAR(200)) Difference
FROM sample_geoms a, sample_geoms b
WHERE a.id = 2 and b.id = 3
```

Risultati:

ID	ID	DIFFERENCE
2	3	POLYGON ((30.00000000 30.00000000, 50.00000000 30.00000000, 50.00000000 40.00000000, 40.00000000 40.00000000, 40.00000000 50.00000000, 30.00000000 50.00000000, 30.00000000 30.00000000))

Esempio 3:

Questo esempio consente di individuare la differenza tra due linee sovrapposte.

```
SELECT a.id, b.id, CAST(ST_AsText(ST_Difference(a.geometry, b.geometry))
as VARCHAR(100)) Difference
FROM sample_geoms a, sample_geoms b
WHERE a.id = 4 and b.id = 5
```

Risultati:

ID	ID	DIFFERENCE
4	5	LINESTRING (70.00000000 70.00000000, 75.00000000 75.00000000)

ST_Dimension

ST_Dimension utilizza una forma geometrica come parametro di input e ne restituisce le dimensioni.

Se la forma geometrica data è vuota, viene restituito -1. Per i punti e i multipunti, la dimensione sarà 0 (zero); per le curve e le multicurve, sarà 1; per i poligoni e i multipoligoni sarà 2. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_Dimension(—forma geometrica—) ◀◀
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry che rappresenta la forma geometrica di cui viene restituita la dimensione.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare diverse forme geometriche e rilevarne le dimensioni.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id char(15), geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
('Empty Point', ST_Geometry('point EMPTY',0))

INSERT INTO sample_geoms VALUES
('Point ZM', ST_Geometry('point zm (10 10 16 30)' ,0))
```

ST_Dimension

```
INSERT INTO sample_geoms VALUES
  ('MultiPoint M', ST_Geometry('multipoint m (10 10 5,
    50 10 6, 10 30 8)', 0))

INSERT INTO sample_geoms VALUES
  ('LineString', ST_Geometry('linestring (10 10, 15 20)', 0))

INSERT INTO sample_geoms VALUES
  ('Polygon', ST_Geometry('polygon((40 120, 90 120, 90 150,
    40 150, 40 120))', 0))

SELECT id, ST_Dimension(geometry) Dimension
FROM sample_geoms
```

Risultati:

ID	DIMENSION
Empty Point	-1
Point ZM	0
MultiPoint M	0
LineString	1
Polygon	2

ST_Disjoint

ST_Disjoint utilizza due forme geometriche come parametri di input e se le due forme non si intersecano, restituisce 1. Se le forme geometriche si intersecano, viene restituito 0 (zero).

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se una delle due forme geometriche è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Disjoint(—forma geometrica 1—, —forma geometrica 2—)
```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry che rappresenta la forma geometrica che verrà separata dalla *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry che rappresenta la forma geometrica che verrà separata dalla *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempi:

Il codice riportato di seguito consente di creare diverse forme geometriche nella tabella SAMPLE_GEOMETRIES.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('polygon((20 30, 30 30, 30 40, 20 40, 20 30))',0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((30 30, 30 50, 50 50, 50 30, 30 30))',0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('polygon((40 40, 40 60, 60 60, 60 40, 40 40))',0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring(60 60, 70 70)' ,0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('linestring(30 30, 40 40)' ,0))
```

Esempio 1:

Questo esempio stabilisce se il primo poligono è disgiunto da una delle forme geometriche.

```
SELECT a.id, b.id, ST_Disjoint(a.geometry, b.geometry) DisJoint
FROM sample_geoms a, sample_geoms b
WHERE a.id = 1
```

Risultati:

ID	ID	DISJOINT
1	1	0
1	2	0
1	3	1
1	4	1
1	5	0

Esempio 2:

Questo esempio stabilisce se il terzo poligono è disgiunto da una delle forme geometriche.

```
SELECT a.id, b.id, ST_Disjoint(a.geometry, b.geometry) DisJoint
FROM sample_geoms a, sample_geoms b
WHERE a.id = 3
```

Risultati:

ID	ID	DISJOINT
3	1	1
3	2	0
3	3	0
3	4	0
3	5	0

Esempio 3:

Questo esempio stabilisce se la seconda linea è disgiunta da una delle forme geometriche.

```
SELECT a.id, b.id, ST_Disjoint(a.geometry, b.geometry) DisJoint
FROM sample_geoms a, sample_geoms b
WHERE a.id = 5
```

ST_Disjoint

Risultati:

ID	ID	DISJOINT
5	1	0
5	2	0
5	3	0
5	4	1
5	5	0

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_Distance

ST_Distance utilizza due forme geometriche ed eventualmente un’unità di misura come parametri di input e restituisce la distanza più breve tra i punti nella prima forma geometrica e quelli nella seconda, calcolata in base all’unità di misura specificata o predefinita.

Per i dati geodetici, ST_Distance restituisce la *distanza geodesica* tra due qualsiasi forme geometriche. La distanza geodesica è la distanza più breve sulla superficie dell’ellissoide. Per ulteriori informazioni, consultare la sezione “Distanze geodesiche” a pagina 160.

Se una delle due forme geometriche è nulla o vuota, viene restituito un valore nullo.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Distance(  
  —forma geometrica 1—, —forma geometrica 2— [  
    —unità di misura— ] )
```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry che rappresenta la forma geometrica da utilizzare per calcolare la distanza dalla *forma geometrica 1*.

forma geometrica 2

Un valore di tipo ST_Geometry che rappresenta la forma geometrica utilizzata per calcolare la distanza dalla *forma geometrica 1*.

unità di misura

Un valore VARCHAR(128) che rappresenta l’unità di misura in cui calcolare il risultato. Le unità di misura supportate sono elencate nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Per i dati geodetici, entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Se il parametro *unità di misura* viene omissso, per determinare l'unità da utilizzare per il risultato vengono applicate le seguenti regole:

- Se la *forma geometrica 1* si trova in un sistema di coordinate proiettate o geocentrico, l'unità di misura lineare associata al sistema sarà quella predefinita.
- Se la *forma geometrica 1* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale, l'unità angolare associata a questo sistema di coordinate sarà utilizzata come predefinita.
- Se la *forma geometrica 1* si trova in un SRS geodetico, l'unità di misura predefinita sarà metri.

Limitazioni sulle conversioni delle unità: se si verifica una delle seguenti condizioni, viene restituito un errore (SQLSTATE 38SU4):

- La forma geometrica si trova in un sistema di coordinate non specificato e viene specificato il parametro *unità*.
- La forma geometrica si trova in un sistema di coordinate proiettate e viene specificata un'unità angolare.
- La forma geometrica si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale geodetico, e viene specificata un'unità lineare.
- La forma geometrica si trova in un sistema di coordinate geografiche, in un SRS geodetico, e viene specificata un'unità angolare.

Tipo restituito:

DOUBLE

Esempi:

Le seguenti istruzioni SQL consentono di creare e riempire le tabelle SAMPLE_GEOMETRIES1 e SAMPLE_GEOMETRIES2.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geometries1(id SMALLINT, spatial_type varchar(13),
  geometry ST_GEOMETRY)
```

```
CREATE TABLE sample_geometries2(id SMALLINT, spatial_type varchar(13),
  geometry ST_GEOMETRY)
```

```
INSERT INTO sample_geometries1(id, spatial_type, geometry)
```

```
VALUES
```

```
( 1, 'ST_Point', ST_Point('point(100 100)', 1) ),
(10, 'ST_LineString', ST_LineString('linestring(125 125, 125 175)', 1) ),
(20, 'ST_Polygon', ST_Polygon('polygon
  ((50 50, 50 150, 150 150, 150 50, 50 50))', 1) )
```

```
INSERT INTO sample_geometries2(id, spatial_type, geometry)
```

```
VALUES
```

```
(101, 'ST_Point', ST_Point('point(200 200)', 1) ),
(102, 'ST_Point', ST_Point('point(200 300)', 1) ),
(103, 'ST_Point', ST_Point('point(200 0)', 1) ),
(110, 'ST_LineString', ST_LineString('linestring(200 100, 200 200)', 1) ),
(120, 'ST_Polygon', ST_Polygon('polygon
  ((200 0, 200 200, 300 200, 300 0, 200 0))', 1) )
```

Esempio 1:

La seguente istruzione SELECT calcola la distanza tra le diverse forme geometriche contenute nelle tabelle SAMPLE_GEOMETRIES1 e SAMPLE_GEOMETRIES2

ST_Distance

```
SELECT  sg1.id AS sg1_id, sg1.spatial_type AS sg1_type,
        sg2.id AS sg1_id, sg2.spatial_type AS sg2_type,
        cast(ST_Distance(sg1.geometry, sg2.geometry)
            AS Decimal(8, 4)) AS distance
FROM    sample_geometries1 sg1, sample_geometries2 sg2
ORDER BY sg1.id
```

Risultati:

SG1_ID	SG1_TYPE	SG1_ID	SG2_TYPE	DISTANCE
1	ST_Point	101	ST_Point	141.4213
1	ST_Point	102	ST_Point	223.6067
1	ST_Point	103	ST_Point	141.4213
1	ST_Point	110	ST_LineString	100.0000
1	ST_Point	120	ST_Polygon	100.0000
10	ST_LineString	101	ST_Point	79.0569
10	ST_LineString	102	ST_Point	145.7737
10	ST_LineString	103	ST_Point	145.7737
10	ST_LineString	110	ST_LineString	75.0000
10	ST_LineString	120	ST_Polygon	75.0000
20	ST_Polygon	101	ST_Point	70.7106
20	ST_Polygon	102	ST_Point	158.1138
20	ST_Polygon	103	ST_Point	70.7106
20	ST_Polygon	110	ST_LineString	50.0000
20	ST_Polygon	120	ST_Polygon	50.0000

Esempio 2:

La seguente istruzione SELECT indica come individuare tutte le forme geometriche la cui distanza una dall'altra non superi 100.

```
SELECT  sg1.id AS sg1_id, sg1.spatial_type AS sg1_type,
        sg2.id AS sg1_id, sg2.spatial_type AS sg2_type,
        cast(ST_Distance(sg1.geometry, sg2.geometry)
            AS Decimal(8, 4)) AS distance
FROM    sample_geometries1 sg1, sample_geometries2 sg2
WHERE   ST_Distance(sg1.geometry, sg2.geometry) <= 100
```

Risultati:

SG1_ID	SG1_TYPE	SG1_ID	SG2_TYPE	DISTANCE
1	ST_Point	110	ST_LineString	100.0000
1	ST_Point	120	ST_Polygon	100.0000
10	ST_LineString	101	ST_Point	79.0569
10	ST_LineString	110	ST_LineString	75.0000
10	ST_LineString	120	ST_Polygon	75.0000
20	ST_Polygon	101	ST_Point	70.7106
20	ST_Polygon	103	ST_Point	70.7106
20	ST_Polygon	110	ST_LineString	50.0000
20	ST_Polygon	120	ST_Polygon	50.0000

Esempio 3:

La seguente istruzione SELECT calcola la distanza in chilometri tra le diverse forme geometriche.

```
Tabelle SAMPLE_GEOMTRIES1 e SAMPLE_GEOMTRIES2.
SELECT  sg1.id AS sg1_id, sg1.spatial_type AS sg1_type,
        sg2.id AS sg1_id, sg2.spatial_type AS sg2_type,
        cast(ST_Distance(sg1.geometry, sg2.geometry, 'KILOMETER')
            AS DECIMAL(10, 4)) AS distance
FROM    sample_geometries1 sg1, sample_geometries2 sg2
ORDER BY sg1.id
```

Risultati:

SG1_ID	SG1_TYPE	SG1_ID	SG2_TYPE	DISTANCE
1	ST_Point	101	ST_Point	12373.2168
1	ST_Point	102	ST_Point	16311.3816
1	ST_Point	103	ST_Point	9809.4713
1	ST_Point	110	ST_LineString	1707.4463
1	ST_Point	120	ST_Polygon	12373.2168
10	ST_LineString	101	ST_Point	8648.2333
10	ST_LineString	102	ST_Point	11317.3934
10	ST_LineString	103	ST_Point	10959.7313
10	ST_LineString	110	ST_LineString	3753.5862
10	ST_LineString	120	ST_Polygon	10891.1254
20	ST_Polygon	101	ST_Point	7700.5333
20	ST_Polygon	102	ST_Point	15039.8109
20	ST_Polygon	103	ST_Point	7284.8552
20	ST_Polygon	110	ST_LineString	6001.8407
20	ST_Polygon	120	ST_Polygon	14515.8872

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_Edge_GC_USA

ST_Edge_GC_USA è la funzione che implementa il DB2SE_USA_GEOCODER utilizzato per eseguire il geocoding degli indirizzi degli Stati Uniti convertendoli in punti. Gli indirizzi vengono confrontati con i file EDGE, forniti sul CD dei dati del geocoder.

La funzione utilizza come parametri di input il nome e il numero della via, il nome della città, lo stato, il codice postale e l'identificativo del sistema di riferimento spaziale del punto ottenuto come risultato, e restituisce un valore ST_Point. Inoltre è possibile specificare diversi parametri di configurazione che possono influenzare il processo di geocoding.

Sintassi:

```

▶▶db2gse.ST_Edge_GC_USA(—via—,—città—,—stato—,—zip—,—srs_id—,——————▶
▶sens_ortografia—,—liv_min_corr—,—offset_laterale—,—unità_offset_laterale—,—offset_finale—,—————▶
▶mappa_base—,—file_indicatore_posizione—)—————▶▶

```

Parametro:

via Un valore di tipo VARCHAR(128) che contiene il numero e il nome della via dell'indirizzo da sottoporre a geocoding.

Il valore non può essere nullo.

città Un valore di tipo VARCHAR(128) che contiene il nome della città dell'indirizzo da sottoporre a geocoding.

Il valore può essere nullo se viene specificato il parametro *zip*.

stato Un valore di tipo VARCHAR(128) che contiene il nome dello stato dell'indirizzo da sottoporre a geocoding. Il nome dello stato può essere abbreviato.

Il valore può essere nullo se viene specificato il parametro *zip*.

- zip* Un valore di tipo VARCHAR(10) che contiene il codice postale dell'indirizzo da sottoporre a geocoding. Il codice postale può essere di cinque cifre o nella forma 5+4.
- Se i parametri *città* e *stato* vengono specificati, il valore può essere nullo.
- srs_id* Un valore di tipo INTEGER che contiene l'identificativo numerico del sistema di riferimento spaziale per il punto ottenuto come risultato. Il valore deve indicare un sistema di riferimento spaziale esistente, che utilizza un sistema di coordinate proiettate basato sul sistema di coordinate geografiche GCS_NORTH_AMERICAN_1983, oppure un sistema di riferimento spaziale che utilizza il sistema di coordinate geografiche GCS_NORTH_AMERICAN_1983.
- Se *srs_id* non identifica uno dei sistemi di riferimento spaziali elencati nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).
- sens_ortografia*
- Un valore di tipo INTEGER che indica la sensibilità ortografica da applicare all'indirizzo dato. Il valore deve essere compreso tra 0 (zero) e 100. Maggiore è il valore, più preciso sarà il geocoder riguardo alle differenze ortografiche dell'indirizzo dato. Le deviazioni risulteranno in un livello di penalità più alto che verrà applicato al valore finale della corrispondenza.
- Se il valore della sensibilità ortografica è troppo alto, solo alcuni indirizzi saranno sottoposti a geocoding con esito positivo, mentre per gli altri verrà restituito un valore nullo. Se la sensibilità ortografica ha un valore troppo basso, verranno considerati validi molti indirizzi non corrispondenti all'input, a causa del livello di differenza accettato nell'ortografia degli indirizzi. **Suggerimento:** Impostare questo valore su 60.
- Se il valore di questo parametro è nullo, la sensibilità ortografica verrà ricavata dal file indicatore di posizione. Se il file non contiene indicazioni, viene utilizzato il valore 60.
- liv_min_corr*
- Un valore di tipo INTEGER che contiene il valore minimo di corrispondenza che indica se un punto corrisponde all'indirizzo dato. Tale valore deve essere compreso tra 0 (zero) e 100. Se il valore del punto è minore del valore *liv_min_corr*, viene restituito un valore nullo e l'indirizzo non viene sottoposto a geocoding.
- Il livello di corrispondenza di un punto dipende da diversi fattori, quali la qualità della carta geografica di base, la sensibilità ortografica o la precisione dell'indirizzo. **Suggerimento:** Impostare questo valore su 80.
- Se il valore di questo parametro è nullo, il livello minimo di corrispondenza verrà ricavato dal file indicatore di posizione. Se il file non contiene indicazioni, viene utilizzato il valore 80.
- offset_laterale*
- Un valore di tipo DOUBLE che indica la distanza che il punto ottenuto come risultato può avere dal punto centrale della strada. Il valore deve essere maggiore o uguale a 0 (zero). Il parametro *unità_offset_laterale* indica l'unità di misura da utilizzare per calcolare l'offset laterale.
- Se il valore di questo parametro è nullo, l'offset laterale verrà ricavato dal file indicatore di posizione. Se il file non contiene indicazioni, viene utilizzato il valore 0,0.

unità_offset_laterale

Un valore di tipo VARCHAR(128) che contiene l'unità di misura in cui viene calcolato il parametro *offset_laterale*. Il valore deve essere una delle seguenti unità:

- Pollice
- Punto
- Piede
- Iarda
- Miglio
- Miglio marinaro
- Millimetro
- Centimetro
- Metro
- Chilometro
- Grado decimale
- Metri proiettati
- Unità dati di riferimento

Se il valore è nullo, le unità di scarto laterale saranno ricavate dal file indicatore di posizione. Se il file non contiene indicazioni, il valore di scarto verrà misurato in piedi.

scarto_finale

Un valore di tipo INTEGER che indica la distanza in cui un punto, che normalmente si troverebbe alla fine del segmento strada, si trova nel segmento. Il valore deve essere maggiore o uguale a 0 (zero). Questo parametro viene utilizzato per evitare di inserire i punti ottenuti come risultato nel centro di una strada all'altezza degli incroci. L'offset finale viene misurato in punti (la risoluzione più piccola fra tutte) sulla carta geografica di base.

Se il valore è nullo, l'offset finale verrà ricavato dal file indicatore di posizione. Se il file non contiene indicazioni, viene utilizzato il valore 3.

mappa_base

Un valore di tipo VARCHAR(256) che contiene il percorso completo, incluso il nome, del file della carta geografica di base (.edg). Tale file è utilizzato dal geocoder per individuare le corrispondenze degli indirizzi dati. E' preferibile utilizzare le carte geografiche di base fornite da DB2 Spatial Extender. E' possibile utilizzare questo parametro se le carte geografiche si trovano in una directory diversa.

Se il valore di questo parametro è nullo, il valore della carta geografica di base verrà ricavato dal file indicatore di posizione. Se il file non contiene indicazioni, la carta verrà ricercata nella directory sqllib dell'istanza corrente, nella sottodirectory *gse/refdata*. Il nome del file cercato è *usa.edg*.

file_indicatore_posizione

Un valore di tipo VARCHAR(256) che rappresenta il percorso completo, incluso il nome, del file indicatore di posizione che contiene ulteriori parametri di configurazione per il geocoder. E' preferibile utilizzare il file indicatore di posizione fornito da DB2 Spatial Extender.

Se il valore è nullo, il file indicatore di posizione verrà ricercato nella directory sqllib dell'istanza corrente, nella sottodirectory *gse/cfg/geocoder*. Il nome del file cercato è *EDGELocator.loc*.

ST_Edge_GC_USA

Tipo restituito:

db2gse.ST_Point

Esempi:

Esempio 1:

Il codice riportato di seguito consente di creare la tabella SAMPLE_GEOCODING e di inserirvi due indirizzi che verranno successivamente sottoposti a geocoding. Il livello minimo di corrispondenza per gli indirizzi dati verrà impostato su 50 e il sistema di riferimento spaziale per i punti ottenuti come risultato è 1.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geocoding (
  street VARCHAR(128),
  city   VARCHAR(128),
  state  VARCHAR(128),
  zip    VARCHAR(5) )

INSERT INTO geocoding(street, city, state, zip)
VALUES ('1212 New York Ave NW', 'Washington', 'DC', '20005'),
('100 First North Street', 'San Jose', 'CA', NULL)

SELECT VARCHAR(ST_AsText(ST_Edge_GC_USA(street, city, state, zip, 1,
  CAST(NULL AS INTEGER), 50, CAST(NULL AS DOUBLE),
  CAST(NULL AS VARCHAR(128)), CAST(NULL AS INTEGER),
  CAST(NULL AS VARCHAR(256)), CAST(NULL AS VARCHAR(256))))), 50)
FROM sample_geocoding
```

Risultati:

```
1
-----
POINT ( -77.02829300 38.90049000)
POINT ( -121.94507200 37.28766700)
```

Esempio 2:

In questo esempio verrà creato un sistema di riferimento spaziale che utilizza un sistema di coordinate proiettate. Per semplificare l'interfaccia della funzione di geocoding, viene creata una funzione definita dall'utente che inglobi la funzione ST_Edge_GC_USA.

```
db2se create_srs <db_name> -srsName CALIFORNIA -srsId 101 -xScale 1
-coordsysName NAD_1983_STATEPLANE_CALIFORNIA_I_FIPS_0401

SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE FUNCTION California_GC (
  street VARCHAR(128), city VARCHAR(128), zip VARCHAR(10))
RETURNS db2gse.ST_Point
LANGUAGE SQL
RETURN db2gse.ST_Edge_GC_USA(street, city, 'CA', zip, 101,
  CAST(NULL AS INTEGER), CAST(NULL AS INTEGER),
  CAST(NULL AS DOUBLE), CAST(NULL AS VARCHAR(128)),
  CAST(NULL AS INTEGER), CAST(NULL AS VARCHAR(256)))

CREATE TABLE sample_geocoding (
  street VARCHAR(128),
  city   VARCHAR(128),
  state  VARCHAR(128),
  zip    VARCHAR(5) )
```

```

INSERT INTO geocoding(street, city, state, zip)
VALUES ('100 First North Street', 'San Jose', 'CA', NULL)

SELECT VARCHAR(ST_AsText(California_GC(street, city, zip)), 50)
FROM sample_geocoding

```

Risultati:

```

1
-----
POINT ( 2004879.00000000 272723.00000000)

NetBIOS

```

Nota: I valori delle coordinate X e Y del punto sono diverse dalle coordinate del primo esempio perché è stato utilizzato un sistema di riferimento diverso.

ST_Endpoint

ST_Endpoint utilizza una curva come parametro di input e restituisce l'ultimo punto della curva. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della curva data.

Se la curva data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```

▶▶ db2gse.ST_EndPoint(—curva—) ▶▶

```

Parametro:

curva Un valore di tipo ST_Curve che rappresenta la forma geometrica di cui viene restituito l'ultimo punto.

Tipo restituito:

db2gse.ST_Point

Esempio:

L'istruzione SELECT consente di individuare il punto finale di ciascuna forma geometrica contenuta nella tabella SAMPLE_LINES.

```

SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines(id INTEGER, line ST_Linestring)

```

```

INSERT INTO sample_lines VALUES
  (1, ST_LineString('linestring (10 10, 5 5, 0 0, 10 0, 5 5, 0 10)', 0) )

INSERT INTO sample_lines VALUES
  (2, ST_LineString('linestring z (0 0 4, 5 5 5, 10 10 6, 5 5 7)', 0) )

```

```

SELECT id, CAST(ST_AsText(ST_EndPoint(line)) as VARCHAR(50)) Endpoint
FROM sample_lines

```

Risultati:

ST_Endpoint

ID	ENDPOINT
1	POINT (0.00000000 10.00000000)
2	POINT Z (5.00000000 5.00000000 7.00000000)

Riferimenti correlati:

- “ST_PointN” a pagina 471

ST_Envelope

ST_Envelope utilizza una forma geometrica come parametro di input ne restituisce l’inviluppo che circonda la forma. L’inviluppo è un rettangolo rappresentato come un poligono.

Se la forma geometrica data è un punto, una linea orizzontale o verticale, viene restituito un rettangolo leggermente più esteso della forma geometrica data. Diversamente, come inviluppo, viene restituito il rettangolo minimo circoscritto alla forma geometrica. Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo. Per restituire l’MBR (minimum bounding rectangle) preciso per tutte le forme geometriche, utilizzare la funzione ST_MBR.

Per i dati geodetici, l’inviluppo è un poligono che racchiude l’MBC della forma geometrica.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_Envelope—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry che rappresenta la forma di cui restituire l’inviluppo.

Tipo restituito:

db2gse.ST_Polygon

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L’esempio consente di creare diverse forme geometriche e di determinare i relativi inviluppi. Per i punti non vuoti e le linee (orizzontali) l’inviluppo risulterà essere un rettangolo leggermente più esteso della forma geometrica.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms VALUES
(1, ST_Geometry('point EMPTY',0))
```

```

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('point zm (10 10 16 30)' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint m (10 10 5, 50 10 6, 10 30 8)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring (10 10, 20 10)',0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('polygon((40 120, 90 120, 90 150, 40 150, 40 120))',0))

SELECT id, CAST(ST_AsText(ST_Envelope(geometry)) as VARCHAR(160)) Envelope
FROM sample_geoms

```

Risultati:

ID	ENVELOPE
1	-
2	POLYGON ((9.00000000 9.00000000, 11.00000000 9.00000000, 11.00000000 11.00000000, 9.00000000 11.00000000, 9.00000000 9.00000000))
3	POLYGON ((10.00000000 10.00000000, 50.00000000 10.00000000, 50.00000000 30.00000000, 10.00000000 30.00000000, 10.00000000 10.00000000))
4	POLYGON ((10.00000000 9.00000000, 20.00000000 9.00000000, 20.00000000 11.00000000, 10.00000000 11.00000000, 10.00000000 9.00000000))
5	POLYGON ((40.00000000 120.00000000, 90.00000000 120.00000000, 90.00000000 150.00000000, 40.00000000 150.00000000, 40.00000000 120.00000000))

Riferimenti correlati:

- “ST_MBR” a pagina 429

ST_EnvIntersects

ST_EnvIntersects utilizza due forme geometriche come parametri di input e se gli involucri delle due forme si intersecano, restituisce 1. In caso contrario, restituisce 0 (zero).

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se una delle forme geometriche date è nulla o vuota, viene restituito un valore nullo.

Sintassi:

```

▶▶db2gse.ST_EnvIntersects(—forma geometrica 1—,—forma geometrica 2—)▶▶

```

Parametro:

ST_EnvIntersects

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'involuppo si interseca con l'involuppo della *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'involuppo si interseca con l'involuppo della *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare due linee parallele e di verificarne l'intersezione. Le linee stesse non si intersecheranno, ma i relativi involuppi sì.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms VALUES
(1, ST_Geometry('linestring (10 10, 50 50)',0))
```

```
INSERT INTO sample_geoms VALUES
(2, ST_Geometry('linestring (10 20, 50 60)',0))
```

```
SELECT a.id, b.id, ST_Intersects(a.geometry, b.geometry) Intersects,
       ST_EnvIntersects(a.geometry, b.geometry) Envelope_Intersects
FROM   sample_geoms a , sample_geoms b
WHERE  a.id = 1 and b.id=2
```

Risultati:

ID	ID	INTERSECTS	ENVELOPE_INTERSECTS
1	2	0	1

ST_EqualCoordsys

ST_EqualCoordsys utilizza due definizioni di sistema di coordinate come parametri di input e se le due definizioni sono identiche, restituisce il valore intero 1 (uno). In caso contrario, restituisce il valore intero 0 (zero). Le definizioni dei sistema di coordinate vengono confrontate a prescindere dalle differenze in spazi, parentesi, caratteri maiuscoli e minuscoli e rappresentazione numeri a virgola mobile.

Se uno dei sistemi di coordinate dati ha valore nullo, viene restituito un valore nullo.

Sintassi:

```
►►—db2gse.ST_EqualCoordsys—(—sistema_coordinate1—,—sistema_coordinate2—)—►►
```

Parametro:

sistema_coordinate1

Un valore di tipo VARCHAR(2048) che definisce il primo sistema di coordinate da paragonare al *sistema_coordinate2*.

sistema_coordinate2

Un valore di tipo VARCHAR(2048) che definisce il secondo sistema di coordinate da paragonare al *sistema_coordinate1*.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di paragonare due sistemi di coordinate australiani e verificare che siano uguali.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
VALUES ST_EqualCoordSys(
  (SELECT definition
   FROM db2gse.ST_COORDINATE_SYSTEMS
   WHERE coordsys_name='GCS_AUSTRALIAN') ,
  (SELECT definition
   FROM db2gse.ST_COORDINATE_SYSTEMS
   WHERE coordsys_name='GCS_AUSTRALIAN_1984')
)
```

Risultati:

```
1
-----
0
```

Riferimenti correlati:

- “Vista del catalogo DB2GSE.ST_COORDINATE_SYSTEMS” a pagina 283

ST_Equals

ST_Equals utilizza due forme geometriche come parametri di input e se le due forme sono uguali, restituisce 1. In caso contrario, restituisce 0 (zero). L'ordine dei punti utilizzati per definire la forma geometrica, non influisce sulla determinazione dell'uguaglianza.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se una delle due forme geometriche date è nulla, viene restituito un valore nullo.

Sintassi:

```
►►—db2gse.ST_Equals—(—forma geometrica 1—,—forma geometrica 2—)—————►►
```

Parametro:

ST_Equals

forma geometrica 1

Un valore di tipo ST_Geometry che rappresenta la forma geometrica che verrà confrontata con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry che rappresenta la forma geometrica che verrà confrontata con la *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempi:

Esempio 1:

Questo esempio consente di creare due poligoni le cui coordinate hanno ordine differente. Per dimostrare che i due poligoni sono considerati uguali, viene utilizzata la funzione ST_Equals.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('polygon((50 30, 30 30, 30 50, 50 50, 50 30))' ,0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((50 30, 50 50, 30 50, 30 30, 50 30))' ,0))

SELECT a.id, b.id, ST_Equals(a.geometry, b.geometry) Equals
FROM sample_geoms a, sample_geoms b
WHERE a.id = 1 and b.id = 2
```

Risultati:

ID	ID	EQUALS
1	2	1

Esempio 2:

Questo esempio consente di creare due forme geometriche con le stesse coordinate X e Y, ma con diverse coordinate M (misure). Quando le forme geometriche vengono paragonate mediante la funzione ST_Equals, per indicare che le forme non sono uguali, viene restituito 0 (zero).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint m(80 80 6, 90 90 7)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('multipoint m(80 80 6, 90 90 4)' ,0))

SELECT a.id, b.id, ST_Equals(a.geometry, b.geometry) Equals
FROM sample_geoms a, sample_geoms b
WHERE a.id = 3 and b.id = 4
```

Risultati:

ID	ID	EQUALS	
	3	4	0

Esempio 3:

Questo esempio consente di creare due forme geometriche aventi una serie differente di coordinate, ma che rappresentano la stessa forma geometrica. ST_Equal confronta le due forme e indica che sono uguali.

```
SET current function path = current function path, db2gse
CREATE TABLE sample_geoms ( id INTEGER, geometry ST_Geometry )
```

```
INSERT INTO sample_geoms VALUES
(5, ST_LineString('linestring ( 10 10, 40 40 )', 0)),
(6, ST_LineString('linestring ( 10 10, 20 20, 40 40)', 0))
```

```
SELECT a.id, b.id, ST_Equals(a.geometry, b.geometry) Equals
FROM sample_geoms a, sample_geoms b
WHERE a.id = 5 AND b.id = 6
```

Risultati:

ID	ID	EQUALS	
	5	6	1

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_EqualsSRS

ST_EqualsSRS utilizza due identificatori di sistema di riferimento spaziale come parametri di input e se i due sistemi sono identici, restituisce valore 1. In caso contrario, restituisce 0 (zero). Vengono confrontati gli offset, i fattori di scala e i sistemi di coordinate.

Se uno degli identificatori dei sistemi di riferimento spaziali ha valore nullo, viene restituito un valore nullo.

Sintassi:

```
►►—db2gse.ST_EqualsSRS—(—srs_id1—,—srs_id2—)—————►►
```

Parametro:

srs_id1 Un valore di tipo INTEGER che identifica il primo sistema di riferimento spaziale da confrontare con il sistema identificato da *srs_id2*.

srs_id2 Un valore di tipo INTEGER che identifica il secondo sistema di riferimento spaziale da confrontare con il sistema identificato da *srs_id1*.

Tipo restituito:

INTEGER

Esempio:

ST_EqualSRS

Utilizzando il seguente richiamo a db2se, vengono creati due sistemi di riferimento spaziali simili.

```
db2se create_srs SAMP_DB -srsId 12 -srsName NYE_12 -xOffset 0 -yOffset 0
-xScale 1 -yScale 1 -coordsysName
NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

```
db2se create_srs SAMP_DB -srsId 22 -srsName NYE_22 -xOffset 0 -yOffset 0
-xScale 1 -yScale 1 -coordsysName
NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

I due SRS hanno gli stessi valori di scarto e di scala e si riferiscono allo stesso sistema di coordinate. L'unica differenza consiste nel nome e nell'ID SRS. Di conseguenza, l'operazione di confronto restituisce 1, che indica che i due sistemi sono gli stessi.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
VALUES ST_EqualSRS(12, 22)
```

Risultati:

```
1
-----
1
```

Riferimenti correlati:

- "Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS" a pagina 291

ST_ExteriorRing

ST_ExteriorRing utilizza un poligono come parametro di input e restituisce la relativa circonferenza esterna come curva. La curva ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale del poligono dato.

Se il poligono dato è nullo o vuoto, viene restituito un valore nullo. Se il poligono non contiene alcuna circonferenza interna, la circonferenza restituita sarà identica ai contorni del poligono.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_ExteriorRing—(—poligono—)—————►►
```

Parametro:

poligono

Un valore di tipo ST_Polygon che rappresenta il poligono di cui restituire la circonferenza esterna.

Tipo restituito:

db2gse.ST_Curve

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio consente di creare due poligoni, uno con circonferenze interne e l'altro senza, quindi di determinare le corrispondenti circonferenze esterne.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

INSERT INTO sample_polys VALUES
  (1, ST_Polygon('polygon((40 120, 90 120, 90 150, 40 150, 40 120),
    (50 130, 60 130, 60 140, 50 140, 50 130),
    (70 130, 80 130, 80 140, 70 140, 70 130))' ,0))

INSERT INTO sample_polys VALUES
  (2, ST_Polygon('polygon((10 10, 50 10, 10 30, 10 10))' ,0))

SELECT id, CAST(ST_AsText(ST_ExteriorRing(geometry))
  AS VARCHAR(180)) Exterior_Ring
FROM sample_polys
```

Risultati:

ID	EXTERIOR_RING
1	LINESTRING (40.00000000 120.00000000, 90.00000000 120.00000000, 90.00000000 150.00000000, 40.00000000 150.00000000, 40.00000000 120.00000000)
2	LINESTRING (10.00000000 10.00000000, 50.00000000 10.00000000, 10.00000000 30.00000000, 10.00000000 10.00000000)

Riferimenti correlati:

- “ST_Boundary” a pagina 352

ST_FindMeasure o ST_LocateAlong

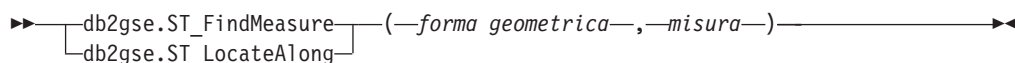
ST_FindMeasure e ST_LocateAlong utilizzano una forma geometrica e una misura come parametri di input e restituiscono un multipunto o una multicurva di quella parte della forma geometrica data che ha esattamente la misura specificata della forma geometrica che contiene la misura specificata. Per i punti e multipunti, vengono restituiti tutti i punti aventi la misura specificata. Per le curve, le multicurve, le superfici e le multisuperfici, per ottenere il risultato viene eseguita una interpolazione. Il calcolo delle superfici e delle multisuperfici viene eseguito in base al contorno della forma geometrica.

Per i punti e multipunti, se non viene individuata la misura specificata, viene restituita una forma geometrica vuota. Per tutte le altre forme geometriche, se la misura data è inferiore alla misura più piccola della forma geometrica o maggiore alla forma più grande, viene restituita una forma geometrica vuota. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

ST_FindMeasure o ST_LocateAlong



Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica in cui ricercare quelle parti con coordinate M (misure) che contengono *misura*.

misura Un valore di tipo DOUBLE che rappresenta la misura delle parti della *forme geometrica* che devono essere incluse nel risultato.

Tipo restituito:

db2gse.ST_Geometry

Esempi:

La seguente istruzione CREATE TABLE consente di creare la tabella SAMPLE_GEOMETRIES. SAMPLE_GEOMETRIES contiene due colonne: la colonna ID, che identifica unicamente ciascuna riga, e la colonna GEOMETRY ST_Geometry, in cui è memorizzata la forma geometrica di esempio.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geometries(id SMALLINT, geometry ST_GEOMETRY)
```

Le seguenti istruzioni INSERT consentono di inserire due righe. La prima corrisponde a una linea; la seconda a un multipunto.

```
INSERT INTO sample_geometries(id, geometry)
VALUES
  (1, ST_LineString('linestring m (2 2 3, 3 5 3, 3 3 6, 4 4 8)', 1)),
  (2, ST_MultiPoint('multipoint m
  (2 2 3, 3 5 3, 3 3 6, 4 4 6, 5 5 6, 6 6 8)', 1))
```

Esempio 1:

Nella seguente istruzione SELECT e nella corrispondente serie di risultati, la funzione ST_FindMeasure viene diretta per individuare i punti la cui misura è 7. La prima riga restituisce un punto. Tuttavia, la seconda riga restituisce un punto vuoto. Per le caratteristiche geografiche lineari (forme geometriche con dimensione maggiore di 0), ST_FindMeasure è in grado di interpolare i punti; tuttavia per i multipunti, è necessario che la misura di destinazione corrisponda esattamente.

```
SELECT id, cast(ST_AsText(ST_FindMeasure(geometry, 7))
  AS varchar(45)) AS measure_7
FROM sample_geometries
```

Risultati:

```
ID      MEASURE_7
-----
1 POINT M ( 3.50000000 3.50000000 7.00000000)
2 POINT EMPTY
```

Esempio 2:

Nella seguente istruzione SELECT e nella corrispondente serie di risultati, la funzione ST_FindMeasure restituisce un punto e un multipunto. La misura di destinazione 6 corrisponde alle misure di ST_FindMeasure e dei dati di origine multipunti.

```
SELECT id, cast(ST_AsText(ST_FindMeasure(geometry, 6))
AS varchar(120)) AS measure_6
FROM sample_geometries
```

Risultati:

ID	MEASURE_6
1	POINT M (3.00000000 3.00000000 6.00000000)
2	MULTIPOINT M (3.00000000 3.00000000 6.00000000, 4.00000000 4.00000000 6.00000000, 5.00000000 5.00000000 6.00000000)

Riferimenti correlati:

- “ST_MeasureBetween, ST_LocateBetween” a pagina 432

ST_Generalize

ST_Generalize utilizza una forma geometrica e una soglia come parametri di input e rappresenta la forma geometrica data con un numero di punti inferiore, mantenendone le caratteristiche generali. Viene utilizzato l’algoritmo di semplificazione linea Douglas-Peucker in base al quale la sequenza di punti che definisce la forma geometrica viene ripetutamente suddivisa fino a quando una serie di punti può essere sostituita da un segmento di una linea retta. Nel segmento lineare, nessuno dei punti definiti devia dalla linea retta più del limite stabilito dalla soglia data. Le coordinate Z ed M non vengono considerate nel processo di semplificazione. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se la forma geometrica è vuota, viene restituita una forma geometrica vuota di tipo ST_Point. Se la forma geometrica o la soglia data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Generalize(—forma geometrica—, —soglia—)
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica a cui viene applicata la semplificazione di linea.

soglia

Un valore di tipo DOUBLE che identifica la soglia da utilizzare per l’algoritmo di semplificazione riga. La soglia dovrà essere maggiore o uguale a 0 (zero). Maggiore è la soglia, minore sarà il numero di punti da utilizzare per rappresentare la geometria generalizzata. Per i dati geodetici, l’unità di misura per la soglia sono i metri.

Tipo restituito:

db2gse.ST_Geometry

Esempi:

ST_Generalize

Nei seguenti esempi, i risultati sono stati formattati nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Viene creata una linea avente otto punti che vanno da (10, 10) a (80, 80). La linea che forma il percorso è quasi retta, fatta eccezione per alcuni punti. Per ridurre il numero di punti nella linea, è possibile utilizzare la funzione ST_Generalize.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, geometry ST_LineString)

INSERT INTO sample_lines VALUES
    (1, ST_LineString('linestring(10 10, 21 20, 34 26, 40 40,
                        52 50, 59 63, 70 71, 80 80)' ,0))
```

Esempio 1:

Utilizzando un fattore di generalizzazione 3, la linea viene ridotta a quattro coordinate e rimane ancora molto simile alla rispettiva rappresentazione originale.

```
SELECT CAST(ST_AsText(ST_Generalize(geometry, 3)) as VARCHAR(115))
    Generalize_3
FROM sample_lines
```

Risultati:

```
GENERALIZE 3
-----
LINESTRING ( 10.00000000 10.00000000, 34.00000000 26.00000000,
            59.00000000 63.00000000, 80.00000000 80.00000000)
```

Esempio 2:

Utilizzando un fattore di generalizzazione 6, la linea viene ridotta a due sole coordinate. Viene prodotta una linea più semplice rispetto alla precedente, ma con più differenze rispetto alla rappresentazione originale.

```
SELECT CAST(ST_AsText(ST_Generalize(geometry, 6)) as VARCHAR(65))
    Generalize_6
FROM sample_lines
```

Risultati:

```
GENERALIZE 6
-----
LINESTRING ( 10.00000000 10.00000000, 80.00000000 80.00000000)
```

ST_GeomCollection

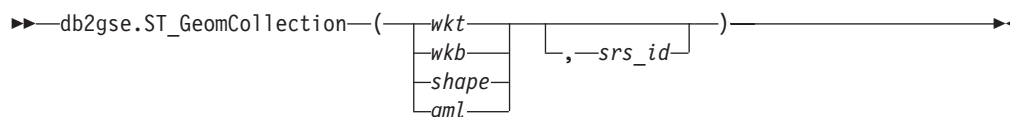
ST_GeomCollection consente di creare un insieme di forme geometriche da uno dei seguenti input:

- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape ESRI
- Una rappresentazione in GML (geography markup language)

Per identificare il sistema di riferimento spaziale in cui verrà inserita l'insieme di forme geometriche ottenuta come risultato, è possibile specificare un identificativo di sistema.

Se le rappresentazioni WKT, WKB, GML o la rappresentazione shape ESRI sono nulle, viene restituito un valore nullo.

Sintassi:



Parametro:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT dell'insieme di forme geometriche ottenuto come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB dell'insieme di forme geometriche ottenuto come risultato.
- shape* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape ESRI dell'insieme di forme geometriche ottenuto come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta l'insieme di forme geometriche ottenuto come risultato utilizzando il linguaggio GML.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per l'insieme di forme geometriche ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_GeomCollection

Note:

Se il parametro *srs_id* viene omesso, potrebbe essere necessario associare *wkt* e *gml* esplicitamente al tipo di dati CLOB. In caso contrario, DB2 potrebbe risolvere con la funzione utilizzata per associare i valori dal tipo di riferimento REF(ST_GeomCollection) al tipo ST_GeomCollection. L'esempio seguente mostra come DB2 con la funzione corretta:

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito mostra come la funzione ST_GeomCollection può essere utilizzata per creare ed inserire un multipunto, una multilinea e un multipoligono dalla rappresentazione WKT e un multipunto da un GML in una colonna GeomCollection.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geomcollections(id INTEGER,
  geometry ST_GEOMCOLLECTION)
```

ST_GeomCollection

```
INSERT INTO sample_geomcollections(id, geometry)
VALUES
(4001, ST_GeomCollection('multipoint(1 2, 4 3, 5 6)', 1) ),
(4002, ST_GeomCollection('multilinestring(
(33 2, 34 3, 35 6),
(28 4, 29 5, 31 8, 43 12),
(39 3, 37 4, 36 7))', 1) ),
(4003, ST_GeomCollection('multipolygon(((3 3, 4 6, 5 3, 3 3),
(8 24, 9 25, 1 28, 8 24),
(13 33, 7 36, 1 40, 10 43, 13 33)))', 1)),
(4004, ST_GeomCollection('<gml:MultiPoint srsName="EPSG:4269"
><gml:PointMember><gml:Point>
<gml:coord><gml:X>10</gml:X>
<gml:Y>20</gml:Y></gml: coord></gml:Point>
</gml:PointMember><gml:PointMember>
<gml:Point><gml:coord><gml:X>30</gml:X>
<gml:Y>40</gml:Y></gml:coord></gml:Point>
</gml:PointMember></gml:MultiPoint>', 1))

SELECT id, cast(geometry..ST_AsText AS varchar(350)) AS geomcollection
FROM sample_geomcollections
```

Risultati:

ID	GEOMCOLLECTION
4001	MULTIPOINT (1.00000000 2.00000000, 4.00000000 3.00000000, 5.00000000 6.00000000)
4002	MULTILINESTRING ((33.00000000 2.00000000, 34.00000000 3.00000000, 35.00000000 6.00000000),(28.00000000 4.00000000, 29.00000000 5.00000000, 31.00000000 8.00000000, 43.00000000 12.00000000),(39.00000000 3.00000000, 37.00000000 4.00000000, 36.00000000 7.00000000))
4003	MULTIPOLYGON (((13.00000000 33.00000000, 10.00000000 43.00000000, 1.00000000 40.00000000, 7.00000000 36.00000000, 13.00000000 33.00000000)),((8.00000000 24.00000000, 9.00000000 25.00000000, 1.00000000 28.00000000, 8.00000000 24.00000000)), ((3.00000000 3.00000000,5.00000000 3.00000000, 4.00000000 6.00000000,3.00000000 3.00000000)))
4004	MULTIPOINT (10.00000000 20.00000000, 30.00000000 40.00000000)

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_GeomCollFromTxt

ST_GeomCollFromTxt utilizza la rappresentazione WKT (well-known text) di un insieme di forme geometriche ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce l’insieme di forme geometriche corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_GeomCollection. La flessibilità di ST_GeomCollection consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
db2gse.ST_GeomCollFromTxt(wkt [, srs_id])
```

Parametro:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT dell'insieme di forme geometriche ottenuto come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per l'insieme di forme geometriche ottenuto come risultato.
- Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).
- Se *srs_id* non identifica uno dei sistemi di riferimento spaziali elencati nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_GeomCollection

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito mostra come la funzione ST_GeomCollFromTxt può essere utilizzata per creare ed inserire un multipunto, una multilinea e un multipoligono da una rappresentazione WKT (well-known text) in una colonna GeomCollection.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geomcollections(id INTEGER, geometry ST_GEOMCOLLECTION)

INSERT INTO sample_geomcollections(id, geometry)
VALUES
  (4011, ST_GeomCollFromTxt('multipoint(1 2, 4 3, 5 6)', 1) ),
  (4012, ST_GeomCollFromTxt('multilinestring(
    (33 2, 34 3, 35 6),
    (28 4, 29 5, 31 8, 43 12),
    (39 3, 37 4, 36 7))', 1) ),
  (4013, ST_GeomCollFromTxt('multipolygon(((3 3, 4 6, 5 3, 3 3),
    (8 24, 9 25, 1 28, 8 24),
    (13 33, 7 36, 1 40, 10 43, 13 33)))', 1))

SELECT id, cast(geometry..ST_AsText AS varchar(340))
       AS geomcollection
FROM   sample_geomcollections
```

Risultati:

ID	GEOMCOLLECTION
4011	MULTIPOINT (1.00000000 2.00000000, 4.00000000 3.00000000,

ST_GeomCollFromTxt

```
5.00000000 6.00000000)
4012      MULTILINESTRING (( 33.00000000 2.00000000, 34.00000000
3.00000000, 35.00000000 6.00000000),( 28.00000000 4.00000000, 29.00000000
5.00000000, 31.00000000 8.00000000, 43.00000000 12.00000000),( 39.00000000
3.00000000, 37.00000000 4.00000000, 36.00000000 7.00000000))
4013      MULTIPOLYGON ((( 13.00000000 33.00000000, 10.00000000 43.00000000,
1.00000000 40.00000000, 7.00000000 36.00000000, 13.00000000 33.00000000)),
(( 8.00000000 24.00000000, 9.00000000 25.00000000, 1.00000000 28.00000000,
8.00000000 24.00000000)),(( 3.00000000 3.00000000, 5.00000000 3.00000000,
4.00000000 6.00000000, 3.00000000 3.00000000)))
```

Riferimenti correlati:

- “ST_GeomCollection” a pagina 388

ST_GeomCollFromWKB

ST_GeomCollFromWKB utilizza la rappresentazione WKB (well-known binary) di un insieme di forme geometriche ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce l'insieme di forme geometriche corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La versione preferita per questa funzionalità è ST_GeomCollection.

Sintassi:

```
►► db2gse.ST_GeomCollFromWKB(—wkb— [—srs_id—]) ◀◀
```

Parametro:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB dell'insieme di forme geometriche ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per l'insieme di forme geometriche ottenuto come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se *srs_id* non identifica uno dei sistemi di riferimento spaziali elencati nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_GeomCollection

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito mostra come la funzione ST_GeomCollFromWKB può essere utilizzata per creare ed interrogare le coordinate di un insieme di forme

geometriche in una rappresentazione WKB. Le righe vengono inserite nella tabella SAMPLE_GEOMCOLLECTION con ID 4021 e 4022 e l'insieme di forme geometriche nel sistema di riferimento spaziale 1.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geomcollections(id INTEGER,
  geometry ST_GEOMCOLLECTION, wkb BLOB(32k))
```

```
INSERT INTO sample_geomcollections(id, geometry)
VALUES
  (4021, ST_GeomCollFromTxt('multipoint(1 2, 4 3, 5 6)', 1)),
  (4022, ST_GeomCollFromTxt('multilinestring(
    (33 2, 34 3, 35 6),
    (28 4, 29 5, 31 8, 43 12))', 1))
```

```
UPDATE sample_geomcollections AS temp_correlated
SET wkb = geometry..ST_AsBinary
WHERE id = temp_correlated.id
```

```
SELECT id, cast(ST_GeomCollFromWKB(wkb)..ST_AsText
  AS varchar(190)) AS GeomCollection
FROM sample_geomcollections
```

Risultati:

```
ID          GEOMCOLLECTION
-----
4021 MULTIPOINT ( 1.00000000 2.00000000, 4.00000000
3.00000000, 5.00000000 6.00000000)

4022 MULTILINESTRING (( 33.00000000 2.00000000,
34.00000000 3.00000000, 35.00000000 6.00000000),( 28.00000000
4.00000000, 29.00000000 5.00000000, 31.00000000 8.00000000,
43.00000000 12.00000000))
```

Riferimenti correlati:

- “Rappresentazione WKB (well-known binary)” a pagina 520

ST_Geometry

ST_Geometry consente di creare una forma geometrica da uno dei seguenti input:

- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape ESRI
- Una rappresentazione in GML (geography markup language)

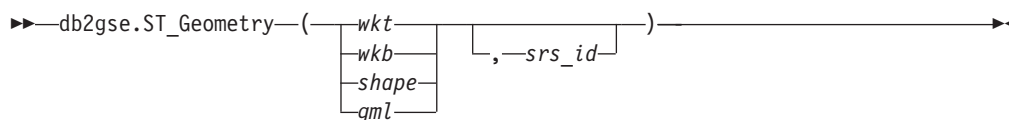
Per identificare il sistema di riferimento spaziale in cui verrà inserita la forma geometrica ottenuta come risultato, è possibile specificare un identificativo di sistema.

Il tipo dinamico della forma geometrica ottenuta come risultato è uno dei tipi secondari istanziabili di ST_Geometry.

Se la rappresentazione WKT, WKB, GML o la rappresentazione shape ESRI sono nulle, viene restituito un valore nullo.

Sintassi:

ST_Geometry



Parametro:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della forma geometrica ottenuta come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della forma geometrica ottenuta come risultato.
- shape* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape ESRI della forma geometrica ottenuta come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta la forma geometrica ottenuta come risultato utilizzando il linguaggio GML.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la forma geometrica ottenuta come risultato.
- Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).
- Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito mostra come la funzione ST_Geometry può essere utilizzata per creare ed inserire un punto da una rappresentazione di punti WKT (well-known text) o una linea da una rappresentazione di linee GML (Geographic Markup Language).

La funzione ST_Geometry è la più flessibile tra le funzioni di costruzione di tipi spaziali, in quanto è in grado di creare qualsiasi tipo spaziale da diverse rappresentazioni di forme geometriche. ST_LineFromText è in grado di creare solo una linea dalla rappresentazione di linee WKT. ST_WKTToSql può costruire qualsiasi tipo, ma solo da una rappresentazione WKT.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geometries(id INTEGER, geometry ST_GEOMETRY)
```

```
INSERT INTO sample_geometries(id, geometry)
VALUES
```

```
(7001, ST_Geometry('point(1 2)', 1) ),
(7002, ST_Geometry('linestring(33 2, 34 3, 35 6)', 1) ),
(7003, ST_Geometry('polygon((3 3, 4 6, 5 3, 3 3))', 1)),
(7004, ST_Geometry('<gml:Point srsName=";EPSG:4269";><gml:coord>
<gml:X>50</gml:X><gml:Y>60</gml:Y></gml:coord>
```

```
</gml:Point>', 1))
```

```
SELECT id, cast(geometry..ST_AsText AS varchar(120)) AS geometry
FROM sample_geometries
```

Risultati:

ID	GEOMETRY
7001	POINT (1.00000000 2.00000000)
7002	LINESTRING (33.00000000 2.00000000, 34.00000000 3.00000000, 35.00000000 6.00000000)
7003	POLYGON ((3.00000000 3.00000000, 5.00000000 3.00000000, 4.00000000 6.00000000, 3.00000000 3.00000000))
7004	POINT (50.00000000 60.00000000)

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515

ST_GeometryN

ST_GeometryN utilizza un insieme di forme geometriche ed un indice come parametri di input e restituisce la forma geometrica nell'insieme che viene identificata dall'indice. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale dell'insieme di forme geometriche dato.

Se l'insieme di forme geometriche ha valore nullo o è vuoto, o se l'indice è minore di 1 o maggiore del numero di forme nell'insieme, viene restituito un valore nullo e viene generata una condizione di avviso (01HS0).

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_GeometryN(—insieme—,—indice—)
```

Parametro:

insieme

Un valore di tipo ST_GeomCollection o uno dei tipi secondari che rappresenta la raccolta di forme geometriche in cui individuare l'*ennesima* forma geometrica.

indice

Un valore di tipo INTEGER che identifica l'*ennesima* forma geometrica restituita dall'*insieme*.

Se l'*indice* è minore di 1 o maggiore del numero di forme geometriche contenute nell'insieme, viene restituito un valore nullo e un messaggio di avviso (SQLSTATE 01HS0).

Tipo restituito:

db2gse.ST_Geometry

Esempio:

ST_GeometryN

Il codice riportato di seguito mostra come scegliere la seconda forma geometrica in un insieme di forme.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geomcollections (id INTEGER,
  geometry ST_GEOMCOLLECTION)

INSERT INTO sample_geomcollections(id, geometry)
VALUES
  (4001, ST_GeomCollection('multipoint(1 2, 4 3)', 1) ),
  (4002, ST_GeomCollection('multilinestring(
    (33 2, 34 3, 35 6),
    (28 4, 29 5, 31 8, 43 12),
    (39 3, 37 4, 36 7))', 1) ),
  (4003, ST_GeomCollection('multipolygon(((3 3, 4 6, 5 3, 3 3),
    (8 24, 9 25, 1 28, 8 24),
    (13 33, 7 36, 1 40, 10 43, 13 33)))', 1))

SELECT id, cast(ST_GeometryN(geometry, 2)..ST_AsText AS varchar(110))
  AS second_geometry
FROM   sample_geomcollections
```

Risultati:

ID	SECOND_GEOMETRY
4001	POINT (4.00000000 3.00000000)
4002	LINestring (28.00000000 4.00000000, 29.00000000 5.00000000, 31.00000000 8.00000000, 43.00000000 12.00000000)
4003	POLYGON ((8.00000000 24.00000000, 9.00000000 25.00000000, 1.00000000 28.00000000, 8.00000000 24.00000000))

Riferimenti correlati:

- “ST_NumGeometries” a pagina 455

ST_GeometryType

ST_GeometryType utilizza una forma geometrica come parametro di input e restituisce il nome completo del tipo dinamico della forma.

Le funzioni DB2 TYPE_SCHEMA e TYPE_NAME producono lo stesso risultato.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_GeometryType—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry per il quale verrà restituito il tipo di forma geometrica.

Tipo restituito:

VARCHAR(128)

Esempi:

Il codice riportato di seguito mostra come determinare il tipo di una geometria.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geometries (id INTEGER, geometry ST_GEOMETRY)

INSERT INTO sample_geometries(id, geometry)
VALUES
  (7101, ST_Geometry('point(1 2)', 1) ),
  (7102, ST_Geometry('linestring(33 2, 34 3, 35 6)', 1) ),
  (7103, ST_Geometry('polygon((3 3, 4 6, 5 3, 3 3))', 1)),
  (7104, ST_Geometry('multipoint(1 2, 4 3)', 1) )

SELECT id, geometry.ST_GeometryType AS geometry_type
FROM   sample_geometries
```

Risultati:

ID	GEOMETRY_TYPE
7101	"DB2GSE"."ST_POINT"
7102	"DB2GSE"."ST_LINESTRING"
7103	"DB2GSE"."ST_POLYGON"
7104	"DB2GSE"."ST_MULTIPPOINT"

ST_GeomFromText

ST_GeomFromText utilizza la rappresentazione WKT (well-known text) di una forma geometrica ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la forma geometrica corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La versione preferita per questa funzionalità è ST_Geometry.

Sintassi:

```
►► db2gse.ST_GeomFromText ( ( wkt ) [ , srs_id ] ) ►►
```

Parametro:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della forma geometrica ottenuta come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la forma geometrica ottenuta come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Geometry

Esempio:

ST_GeomFromText

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

In questo esempio, la funzione ST_GeomFromText viene utilizzata per creare ed inserire un punto da una rappresentazione WKT (well known text).

Il codice riportato di seguito consente di inserire righe nella tabella SAMPLE_POINTS aventi ID e forme geometriche nel sistema di riferimento spaziale 1, utilizzando la rappresentazione WKT.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geometries(id INTEGER, geometry ST_GEOMETRY)

INSERT INTO sample_geometries(id, geometry)
VALUES
  (1251, ST_GeomFromText('point(1 2)', 1) ),
  (1252, ST_GeomFromText('linestring(33 2, 34 3, 35 6)', 1) ),
  (1253, ST_GeomFromText('polygon((3 3, 4 6, 5 3, 3 3))', 1))
```

La seguente istruzione SELECT restituirà l'ID e le forme geometriche dalla tabella SAMPLE_GEOMETRIES.

```
SELECT id, cast(geometry..ST_AsText AS varchar(105))
       AS geometry
FROM   sample_geometries
```

Risultati:

ID	GEOMETRY
1251	POINT (1.00000000 2.00000000)
1252	LINestring (33.00000000 2.00000000, 34.00000000 3.00000000, 35.00000000 6.00000000)
1253	POLYGON ((3.00000000 3.00000000, 5.00000000 3.00000000, 4.00000000 6.00000000, 3.00000000 3.00000000))

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515

ST_GeomFromWKB

ST_GeomFromWKB utilizza la rappresentazione WKB (well-known binary) di una forma geometrica ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la forma geometrica corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La versione preferita per questa funzionalità è ST_Geometry.

Sintassi:

```
db2gse.ST_GeomFromWKB(wkb, srs_id)
```

Parametro:

- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della forma geometrica ottenuta come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la forma geometrica ottenuta come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Geometry

Esempi:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito mostra come la funzione ST_GeomFromWKB può essere utilizzata per creare ed inserire una linea da una rappresentazione di linea WKB (well-known binary).

L'esempio seguente inserisce un record con ID nella tabella SAMPLE_GEOMETRIES e una forma geometrica nel sistema di riferimento spaziale 1, in una rappresentazione WKB.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geometries (id INTEGER, geometry ST_GEOMETRY,
                                wkb BLOB(32K))

INSERT INTO sample_geometries(id, geometry)
VALUES
    (1901, ST_GeomFromText('point(1 2)', 1) ),
    (1902, ST_GeomFromText('linestring(33 2, 34 3, 35 6)', 1) ),
    (1903, ST_GeomFromText('polygon((3 3, 4 6, 5 3, 3 3))', 1))

UPDATE sample_geometries AS temp_correlated
SET    wkb = geometry..ST_AsBinary
WHERE id = temp_correlated.id

SELECT id, cast(ST_GeomFromWKB(wkb)..ST_AsText AS varchar(190))
       AS geometry
FROM   sample_geometries
```

Risultati:

ID	GEOMETRY
1901	POINT (1.00000000 2.00000000)
1902	LINestring (33.00000000 2.00000000, 34.00000000 3.00000000, 35.00000000 6.00000000)
1903	POLYGON ((3.00000000 3.00000000, 5.00000000 3.00000000, 4.00000000 6.00000000, 3.00000000 3.00000000))

Riferimenti correlati:

- “Rappresentazione WKB (well-known binary)” a pagina 520

ST_GetIndexParms

ST_GetIndexParms utilizza l'identificativo di un indice spaziale o di una colonna spaziale come parametro di input e restituisce i parametri utilizzati per definire l'indice o l'indice della colonna spaziale. Se viene specificato un ulteriore numero come parametro, viene restituita solo la dimensione della griglia identificata dal numero.

Sintassi:

```
db2gse.ST_GetIndexParms(
  schema_indice, nome_indice,
  schema_tabella, nome_tabella, nome_colonna,
  numero_dimens_griglia
)
```

Parametro:

schema_indice

Un valore di tipo VARCHAR(128) che identifica lo schema in cui si trova l'indice spaziale *nome_indice*. Il nome dello schema rileva la distinzione tra maiuscolo e minuscolo e deve essere riportato nella vista del catalogo SYSCAT.SCHEMATA.

Se il parametro è nullo, come nome dello schema per l'indice spaziale viene utilizzato il valore del registro speciale CURRENT SCHEMA.

nome_indice

Un valore di tipo VARCHAR(128) che contiene il nome non qualificato dell'indice spaziale per il quale vengono restituiti i parametri di indice. Il nome dell'indice rileva la distinzione tra maiuscolo e minuscolo e deve essere riportato nella vista del catalogo SYSCAT.INDEXES per lo schema *schema_indice*.

schema_tabella

Un valore di tipo VARCHAR(128) che identifica lo schema in cui si trova la tabella *nome_tabella*. Il nome dello schema rileva la distinzione tra maiuscolo e minuscolo e deve essere riportato nella vista del catalogo SYSCAT.SCHEMATA.

Se il parametro è nullo, come nome dello schema per l'indice spaziale viene utilizzato il valore del registro speciale CURRENT SCHEMA.

nome_tabella

Un valore di tipo VARCHAR(128) che contiene il nome della tabella contenente la colonna spaziale *nome_colonna*. Il nome della tabella rileva la distinzione tra maiuscolo e minuscolo e deve essere riportato nella vista del catalogo SYSCAT.TABLES per lo schema *schema_tabella*.

nome_colonna

Un valore di tipo VARCHAR(128) che identifica la colonna nella tabella *schema_tabella.nome_tabella* per la quale vengono restituiti i parametri dell'indice spaziale della colonna. Il nome della colonna rileva la

distinzione tra maiuscolo e minuscolo e deve essere riportata nella vista del catalogo SYSCAT.COLUMNS per la tabella *schema_tabella.nome_tabella*.

Se nella colonna non è stato definito alcun indice spaziale, viene generato un errore (SQLSTATE 38SQ0).

numero_dimens_griglia

Un valore DOUBLE che identifica il parametro di cui viene restituito il valore o i valori.

Se il valore è minore di 1 o maggiore di 3, viene generato un errore (SQLSTATE 38SQ1).

Tipo restituito:

DOUBLE (se viene specificato il parametro *numero_dimens_griglia*)

Se il parametro *numero_dimens_griglia* non viene specificato, viene restituita una tabella contenente le colonne ORDINAL e VALUE. La colonna ORDINAL è di tipo INTEGER, mentre la colonna VALUE è di tipo DOUBLE.

Se i parametri restituiti sono rivolti ad un indice di griglia, la colonna ORDINAL conterrà i valori 1, 2 e 3 rispettivamente per la prima, la seconda e la terza dimensione griglia. La colonna VALUE conterrà le dimensioni della griglia.

La colonna VALUE contiene i valori di ciascun parametro.

Esempi:

Il codice riportato di seguito consente di creare una tabella contenente una colonna e un indice spaziali.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sch.offices (name VARCHAR(30), location ST_Point )
```

```
CREATE INDEX sch.idx ON sch.offices(location)
EXTEND USING db2gse.spatial_index(1e0, 10e0, 1000e0)
```

Per richiamare i valori dei parametri utilizzati durante la creazione dell'indice spaziale, è possibile utilizzare la funzione ST_GetIndexParms.

Esempio 1:

Questo esempio indica come richiamare le tre dimensioni della griglia di un indice spaziale separatamente, indicando esplicitamente quali parametri, identificati dai rispettivi numeri, dovranno essere restituiti.

```
VALUES ST_GetIndexParms('SCH', 'OFFICES', 'LOCATION', 1)
```

Risultati:

```
1
-----
+1.000000000000000E+000
```

```
VALUES ST_GetIndexParms('SCH', 'OFFICES', 'LOCATION', 2)
```

Risultati:

```
1
-----
+1.000000000000000E+001
```

ST_GetIndexParms

```
VALUES ST_GetIndexParms('SCH', 'IDX', 3)
```

Risultati:

```
1
-----
+1.000000000000000E+003
```

Esempio 2:

L'esempio riportato di seguito mostra come richiamare tutti i parametri di un indice di griglia spaziale. La funzione ST_GetIndexParms restituisce una tabella che indica il numero del parametro e la dimensione della griglia corrispondente.

```
SELECT * FROM TABLE ( ST_GetIndexParms('SCH', 'OFFICES', 'LOCATION') ) AS t
```

Risultati:

ORDINAL	VALUE
1	+1.000000000000000E+000
2	+1.000000000000000E+001
3	+1.000000000000000E+003

```
SELECT * FROM TABLE ( ST_GetIndexParms('SCH', 'IDX') ) AS t
```

Risultati:

ORDINAL	VALUE
1	+1.000000000000000E+000
2	+1.000000000000000E+001
3	+1.000000000000000E+003

Argomenti correlati:

- "Indici di griglia spaziali" a pagina 100

ST_InteriorRingN

ST_InteriorRingN utilizza un poligono ed un indice come parametri di input e restituisce la circonferenza interna identificata dall'indice dato, sotto forma di linea. Le circonferenze interne sono organizzate in base alle regole definite dalle routine di verifica delle geometrie interne.

Se il poligono dato ha valore nullo o è vuoto, o se non contiene alcuna circonferenza, viene restituito un valore nullo. Se l'indice è minore di 1 o maggiore del numero di circonferenze interne nel poligono, viene restituito un valore nullo e viene generata una condizione di avviso (IHS1).

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_InteriorRingN(poligono, indice)
```

Parametro:

poligono

Un valore di tipo ST_Polygon che rappresenta la forma geometrica in base alla quale viene restituita la circonferenza interna identificata dall'*indice*.

indice Un valore di tipo INTEGER che identifica l'*ennesima* circonferenza interna. Se l'*indice* non individua alcuna circonferenza interna, viene generata una condizione di avviso (01HS1).

Tipo restituito:

db2gse.ST_Curve

Esempio:

Questo esempio consente di creare un poligono con due circonferenze interne. Per creare la seconda circonferenza, richiamare la funzione ST_InteriorRingN.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

INSERT INTO sample_polys VALUES
  (1, ST_Polygon('polygon((40 120, 90 120, 90 150, 40 150, 40 120),
    (50 130, 60 130, 60 140, 50 140, 50 130),
    (70 130, 80 130, 80 140, 70 140, 70 130))' ,0))

SELECT id, CAST(ST_AsText(ST_InteriorRingN(geometry, 2)) as VARCHAR(180))
       Interior_Ring
FROM sample_polys
```

Risultati:

```
ID          INTERIOR_RING
-----
1 LINESTRING ( 70.00000000 130.00000000, 70.00000000 140.00000000,
80.00000000 140.00000000, 80.00000000 130.00000000, 70.00000000 130.00000000)
```

Riferimenti correlati:

- “ST_ExteriorRing” a pagina 384
- “ST_NumInteriorRing” a pagina 456

ST_Intersection

ST_Intersection utilizza due forme geometriche come parametri di input e restituisce la forma geometrica corrispondente alla intersezione tra le forme date. L'intersezione è la parte comune della prima e della seconda forma geometrica. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della prima forma geometrica.

Se possibile, il tipo specifico della forma geometrica restituita sarà ST_Point, ST_LineString o ST_Polygon. Ad esempio, l'intersezione di un punto e un poligono è vuota o è costituita da un singolo punto, rappresentato come ST_MultiPoint.

Se una delle due forme geometriche è nulla, viene restituito un valore nullo.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

La funzione può essere richiamata come metodo.

ST_Intersection

Sintassi:

```
►► db2gse.ST_Intersection(—forma geometrica 1—,—forma geometrica 2—)◄◄
```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la prima forma geometrica di cui calcolare l'intersezione con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la seconda forma geometrica di cui calcolare l'intersezione con la *forma geometrica 1*.

Per i dati geodetici, entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Tipo restituito:

db2gse.ST_Geometry

La dimensione della forma geometrica restituita corrisponde a quella dell'input di dimensioni inferiori, ad eccezione delle linee nei dati geodetici. Per i dati geodetici, la dimensione dell'intersezione di due linee è 0 (in altre parole, l'intersezione è un punto o un multipunto).

Esempio:

Nei seguenti esempi, i risultati sono stati formattati nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio consente di creare diverse forme geometriche e di determinare un'eventuale intersezione con la prima forma creata.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('polygon((30 30, 30 50, 50 50, 50 30, 30 30))' ,0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((20 30, 30 30, 30 40, 20 40, 20 30))' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('polygon((40 40, 40 60, 60 60, 60 40, 40 40))' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring(60 60, 70 70)' ,0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('linestring(30 30, 60 60)' ,0))

SELECT a.id, b.id, CAST(ST_AsText(ST_Intersection(a.geometry, b.geometry))
  as VARCHAR(150)) Intersection
FROM sample_geoms a, sample_geoms b
WHERE a.id = 1
```

Risultati:

```

ID          ID          INTERSECTION
-----
          1          1 POLYGON (( 30.00000000 30.00000000, 50.00000000
30.00000000, 50.00000000 50.00000000, 30.00000000 50.00000000, 30.00000000
30.00000000))

          1          2 LINESTRING ( 30.00000000 40.00000000, 30.00000000
30.00000000)

          1          3 POLYGON (( 40.00000000 40.00000000, 50.00000000
40.00000000, 50.00000000 50.00000000, 40.00000000 50.00000000, 40.00000000
40.00000000))

          1          4 POINT EMPTY

          1          5 LINESTRING ( 30.00000000 30.00000000, 50.00000000
50.00000000)

5 record selezionati.

```

ST_Intersects

ST_Intersects utilizza due forme geometriche come parametri di input e se le due forme si intersecano, restituisce 1. Se le forme geometriche non si intersecano, viene restituito 0 (zero).

Se una delle due forme geometriche è nulla o vuota, viene restituito un valore nullo.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

Sintassi:

```

▶▶—db2gse.ST_Intersects—(—forma geometrica 1—,—forma geometrica 2—)—▶▶

```

Parametro:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui calcolare l'intersezione con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui calcolare l'intersezione con la *forma geometrica 1*.

Limitazioni: per i dati geodetici, entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Tipo restituito:

INTEGER

Esempio:

Le seguenti istruzioni consentono di creare e riempire le tabelle SAMPLE_GEOMETRIES1 e SAMPLE_GEOMETRIES2.

ST_Intersects

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_geometries1(id SMALLINT, spatial_type varchar(13),
  geometry ST_GEOMETRY);
CREATE TABLE sample_geometries2(id SMALLINT, spatial_type varchar(13),
  geometry ST_GEOMETRY);

INSERT INTO sample_geometries1(id, spatial_type, geometry)
VALUES
  ( 1, 'ST_Point', ST_Point('point(550 150)', 1) ),
  (10, 'ST_LineString', ST_LineString('linestring(800 800, 900 800)', 1)),
  (20, 'ST_Polygon', ST_Polygon('polygon((500 100, 500 200, 700 200,
    700 100, 500 100))', 1) )

INSERT INTO sample_geometries2(id, spatial_type, geometry)
VALUES
  (101, 'ST_Point', ST_Point('point(550 150)', 1) ),
  (102, 'ST_Point', ST_Point('point(650 200)', 1) ),
  (103, 'ST_Point', ST_Point('point(800 800)', 1) ),
  (110, 'ST_LineString', ST_LineString('linestring(850 250, 850 850)', 1)),
  (120, 'ST_Polygon', ST_Polygon('polygon((650 50, 650 150, 800 150,
    800 50, 650 50))', 1)),
  (121, 'ST_Polygon', ST_Polygon('polygon((20 20, 20 40, 40 40, 40 20,
    20 20))', 1) )
```

La seguente istruzione SELECT consente di determinare se le forme geometriche contenute nelle tabelle SAMPLE_GEOMETRIES1 e SAMPLE_GEOMETRIES2 si intersecano.

```
SELECT  sg1.id AS sg1_id, sg1.spatial_type AS sg1_type,
        sg2.id AS sg2_id, sg2.spatial_type AS sg2_type,
        CASE ST_Intersects(sg1.geometry, sg2.geometry)
          WHEN 0 THEN 'Geometries do not intersect'
          WHEN 1 THEN 'Geometries intersect'
        END AS intersects
FROM    sample_geometries1 sg1, sample_geometries2 sg2
ORDER BY sg1.id
```

Risultati:

SG1_ID	SG1_TYPE	SG2_ID	SG2_TYPE	INTERSECTS
1	ST_Point	101	ST_Point	Geometries intersect
1	ST_Point	102	ST_Point	Geometries do not intersect
1	ST_Point	103	ST_Point	Geometries do not intersect
1	ST_Point	110	ST_LineString	Geometries do not intersect
1	ST_Point	120	ST_Polygon	Geometries do not intersect
1	ST_Point	121	ST_Polygon	Geometries do not intersect
10	ST_LineString	101	ST_Point	Geometries do not intersect
10	ST_LineString	102	ST_Point	Geometries do not intersect
10	ST_LineString	103	ST_Point	Geometries intersect
10	ST_LineString	110	ST_LineString	Geometries intersect
10	ST_LineString	120	ST_Polygon	Geometries do not intersect
10	ST_LineString	121	ST_Polygon	Geometries do not intersect
20	ST_Polygon	101	ST_Point	Geometries intersect
20	ST_Polygon	102	ST_Point	Geometries intersect
20	ST_Polygon	103	ST_Point	Geometries do not intersect
20	ST_Polygon	110	ST_LineString	Geometries do not intersect
20	ST_Polygon	120	ST_Polygon	Geometries intersect
20	ST_Polygon	121	ST_Polygon	Geometries do not intersect

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_Is3d

ST_Is3d utilizza una forma geometrica come parametro di input e restituisce 1 se rileva nella forma le coordinate Z. In caso contrario, restituisce 0 (zero).

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_Is3D—(*—forma geometrica—*)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per rilevare la presenza delle coordinate Z.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare diverse forme geometriche con e senza coordinate Z ed M (misure). Successivamente, per determinare quale forma contiene coordinate Z, viene utilizzata la funzione ST_Is3d.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('point EMPTY',0))
```

```
INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((40 120, 90 120, 90 150, 40 150, 40 120))' ,0))
```

```
INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint m (10 10 5, 50 10 6, 10 30 8)' ,0))
```

```
INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring z (10 10 166, 20 10 168)',0))
```

```
INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('point zm (10 10 16 30)' ,0))
```

```
SELECT id, ST_Is3d(geometry) Is_3D
FROM sample_geoms
```

Risultati:

ID	IS_3D
1	0
2	0
3	0
4	1

ST_IsClosed

ST_IsClosed utilizza una curva o una multicurva come parametro di input e se una delle due risulta chiusa, restituisce 1. In caso contrario, restituisce 0 (zero).

Una curva è definita chiusa, quando il punto di iniziale e quello finale coincidono. Se la curva ha coordinate Z, le coordinate Z dei punti iniziale e finale, devono essere uguali. In caso contrario, i punti non verranno considerati uguali, quindi la curva non sarà ritenuta chiusa. Una multicurva è chiusa se tutte le curve che la compongono sono chiuse.

Se la curva o la multicurva è vuota, viene restituito 0 (zero). Se nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►► db2gse.ST_IsClosed(—*curva*—)◄◄

Parametro:

curva Un valore di tipo ST_Curve o ST_MultiCurve o uno dei tipi secondari, che rappresenta la curva o la multicurva da analizzare.

Tipo restituito:

INTEGER

Esempi:

Esempio 1:

Questo esempio consente di creare diverse linee. Le ultime due linee avranno le stesse coordinate X e Y, ma una linea conterrà diverse coordinate Z per cui la linea non sarà chiusa, e l'altra conterrà diverse coordinate M (misure) che non interferiranno sulla chiusura della linea.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, geometry ST_Linestring)

INSERT INTO sample_lines VALUES
  (1, ST_Linestring('linestring EMPTY',0))

INSERT INTO sample_lines VALUES
  (2, ST_Linestring('linestring(10 10, 20 10, 20 20)' ,0))

INSERT INTO sample_lines VALUES
  (3, ST_Linestring('linestring(10 10, 20 10, 20 20, 10 10)' ,0))

INSERT INTO sample_lines VALUES
  (4, ST_Linestring('linestring m(10 10 1, 20 10 2, 20 20 3,
  10 10 4)' ,0))

INSERT INTO sample_lines VALUES
```

```
(5, ST_Linestring('linestring z(10 10 5, 20 10 6, 20 20 7,
10 10 8)' ,0))
```

```
SELECT id, ST_IsClosed(geometry) Is_Closed
FROM sample_lines
```

Risultati:

ID	IS_CLOSED
1	0
2	0
3	1
4	1
5	0

Esempio 2:

Questo esempio consente di creare due multilinee. Per determinare se le multilinee sono chiuse, viene utilizzata la funzione ST_IsClosed. La prima multilinea non risulterà chiusa, anche se tutte le curve insieme formano un cerchio completo. Il risultato si deve al fatto che le curve, considerate singolarmente, non sono chiuse.

La seconda multilinea è chiusa perché le curve che la compongono sono chiuse.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mlines (id INTEGER, geometry ST_MultiLinestring)
INSERT INTO sample_mlines VALUES
(6, ST_MultiLinestring('multilinestring((10 10, 20 10, 20 20),
(20 20, 30 20, 30 30),
(30 30, 10 30, 10 10))',0))
```

```
INSERT INTO sample_mlines VALUES
(7, ST_MultiLinestring('multilinestring((10 10, 20 10, 20 20, 10 10 ),
(30 30, 50 30, 50 50,
30 30 ))',0))
```

```
SELECT id, ST_IsClosed(geometry) Is_Closed
FROM sample_mlines
```

Risultati:

ID	IS_CLOSED
6	0
7	1

ST_IsEmpty

ST_IsEmpty utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è vuota. In caso contrario, restituisce 0 (zero). Una forma geometrica è vuota se non dispone di punti che la definiscono.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

ST_IsEmpty

►►—db2gse.ST_IsEmpty—(*—forma geometrica—*)—►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare.

Tipo restituito:

INTEGER

Esempio:

Il codice riportato di seguito consente di creare tre forme geometriche e di determinare se sono vuote.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('point EMPTY',0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((40 120, 90 120, 90 150, 40 150, 40 120))' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint m (10 10 5, 50 10 6, 10 30 8)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring z (10 10 166, 20 10 168)',0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('point zm (10 10 16 30)' ,0))

SELECT id, ST_IsEmpty(geometry) Is_Empty
FROM sample_geoms
```

Risultati:

ID	IS_EMPTY
1	1
2	0
3	0
4	0
5	0

ST_IsMeasured

ST_IsMeasured utilizza una forma geometrica come parametro di input e restituisce 1 se rileva nella forma le coordinate M (misure). In caso contrario, restituisce 0 (zero).

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►—db2gse.ST_IsMeasured—(*—forma geometrica—*)—►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per rilevare la presenza delle coordinate M (misure).

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare diverse forme geometriche con e senza coordinate Z ed M (misure). Successivamente, per determinare quale forma contiene delle misure, viene utilizzata la funzione ST_IsMeasured.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('point EMPTY',0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('polygon((40 120, 90 120, 90 150, 40 150, 40 120))' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint m (10 10 5, 50 10 6, 10 30 8)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('linestring z (10 10 166, 20 10 168)',0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('point zm (10 10 16 30)' ,0))

SELECT id, ST_IsMeasured(geometry) Is_Measured
FROM sample_geoms
```

Risultati:

ID	IS_MEASURED
1	0
2	0
3	1
4	0
5	1

ST_IsRing

ST_IsRing utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è una circonferenza. In caso contrario, restituisce 0 (zero). Una curva è definita una circonferenza se è semplice e chiusa.

Se la curva è vuota, viene restituito 0 (zero). Se nulla, viene restituito un valore nullo.

ST_IsRing

La funzione può essere richiamata come metodo.

Sintassi:

► db2gse.ST_IsRing(*—curva—*) ◀

Parametro:

curva Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la curva da analizzare.

Tipo restituito:

INTEGER

Esempi:

Questo esempio consente di creare quattro linee. Per controllare se le linee formano delle circonferenze, viene utilizzata la funzione ST_IsRing. L'ultima linea, anche se chiusa, non viene considerata una circonferenza perché è attraversata dal percorso.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, geometry ST_Linestring)

INSERT INTO sample_lines VALUES
  (1, ST_Linestring('linestring EMPTY',0))

INSERT INTO sample_lines VALUES
  (2, ST_Linestring('linestring(10 10, 20 10, 20 20)' ,0))

INSERT INTO sample_lines VALUES
  (3, ST_Linestring('linestring(10 10, 20 10, 20 20, 10 10)' ,0))

INSERT INTO sample_lines VALUES
  (4, ST_Linestring('linestring(10 10, 20 10, 10 20, 20 20, 10 10)' ,0))

SELECT id, ST_IsClosed(geometry) Is_Closed, ST_IsRing(geometry) Is_Ring
FROM sample_lines
```

Risultati:

ID	IS_CLOSED	IS_RING
1	1	0
2	0	0
3	1	1
4	1	0

Riferimenti correlati:

- "ST_IsClosed" a pagina 408
- "ST_IsSimple" a pagina 412

ST_IsSimple

ST_IsSimple utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è semplice. In caso contrario, restituisce 0 (zero).

I punti, le superfici e le multisuperfici sono sempre forme di tipo semplice. Una curva è semplice se non passa due volte per lo stesso punto; un multipunto è semplice se non contiene due punti uguali; una multicurva è semplice se tutte le curve che la costituiscono sono semplici e le uniche intersezioni presenti sono quelle dei contorni delle curve all'interno della multicurva.

Se la forma geometrica data è vuota, viene restituito 1. Se nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_IsSimple—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare.

Tipo restituito:

INTEGER

Esempi:

Questo esempio consente di creare diverse forme geometriche e di controllare se le forme create sono semplici. La forma geometrica con ID 4, non è considerata semplice perché contiene più volte lo stesso punto. La forma geometrica con ID 6 non è considerata semplice perché è attraversata dalla linea.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms VALUES
  (1, ST_Geometry('point EMPTY' ,0))

INSERT INTO sample_geoms VALUES
  (2, ST_Geometry('point (21 33)' ,0))

INSERT INTO sample_geoms VALUES
  (3, ST_Geometry('multipoint(10 10, 20 20, 30 30)' ,0))

INSERT INTO sample_geoms VALUES
  (4, ST_Geometry('multipoint(10 10, 20 20, 30 30, 20 20)' ,0))

INSERT INTO sample_geoms VALUES
  (5, ST_Geometry('linestring(60 60, 70 60, 70 70)' ,0))

INSERT INTO sample_geoms VALUES
  (6, ST_Geometry('linestring(20 20, 30 30, 30 20, 20 30 )' ,0))

INSERT INTO sample_geoms VALUES
  (7, ST_Geometry('polygon((40 40, 50 40, 50 50, 40 40 ))' ,0))

SELECT id, ST_IsSimple(geometry) Is_Simple
FROM sample_geoms
```

Risultati:

ST_IsSimple

ID	IS_SIMPLE
1	1
2	1
3	1
4	0
5	1
6	0
7	1

ST_IsValid

ST_IsValid utilizza una forma geometrica come parametro di input e restituisce 1 se rileva che la forma è valida. In caso contrario, restituisce 0 (zero). Una forma geometrica è valida solo se tutti gli attributi nel tipo strutturato sono coerenti con la rappresentazione interna dei dati geometrici, e se la rappresentazione non è danneggiata.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►—db2gse.ST_IsValid—(*—forma geometrica—*)—►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari.

Tipo restituito:

INTEGER

Esempio:

Questo esempio consente di creare diverse forme geometriche e di utilizzare ST_IsValid per verificare che le forme siano valide. Tutte le forme geometriche saranno valide perché le routine di costruzione, ad esempio ST_Geometry, non consentono di creare forme geometriche non valide.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms VALUES
(1, ST_Geometry('point EMPTY',0))
```

```
INSERT INTO sample_geoms VALUES
(2, ST_Geometry('polygon((40 120, 90 120, 90 150, 40 150, 40 120))' ,0))
```

```
INSERT INTO sample_geoms VALUES
(3, ST_Geometry('multipoint m (10 10 5, 50 10 6, 10 30 8)' ,0))
```

```
INSERT INTO sample_geoms VALUES
(4, ST_Geometry('linestring z (10 10 166, 20 10 168)',0))
```

```
INSERT INTO sample_geoms VALUES
```



```
(5, ST_Geometry('point zm (10 10 16 30)' ,0))
```

```
SELECT id, ST_IsValid(geometry) Is_Valid
FROM sample_geoms
```

Risultati:

ID	IS_VALID
1	1
2	1
3	1
4	1
5	1

ST_Length

ST_Length utilizza una curva o una multicurva ed, eventualmente, un'unità di misura come parametri di input e restituisce la lunghezza della curva o multicurva data nell'unità di misura predefinita o specificata.

Se la curva o multicurva data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Length(curva [, unità di misura])
```

Parametro:

curva Un valore di tipo ST_Curve o ST_MultiCurve che rappresenta le curve di cui viene restituita la lunghezza.

unità di misura

Un valore VARCHAR(128) che rappresenta l'unità di misura in cui calcolare la lunghezza della curva. Le unità di misura supportate sono elencate nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Se il parametro *unità di misura* viene omesso, per determinare l'unità da utilizzare per misurare la lunghezza, vengono applicate le seguenti regole:

- Se la *curva* si trova in un sistema di coordinate proiettate o geocentrico, l'unità di misura lineare associata al sistema sarà quella predefinita.
- Se la *curva* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale, l'unità angolare associata a questo sistema di coordinate sarà utilizzata come predefinita.
- Se la *curva* si trova in un SRS geodetico, l'unità di misura predefinita sarà metri.

Limitazioni sulle conversioni delle unità: se si verifica una delle seguenti condizioni, viene restituito un errore (SQLSTATE 38SU4):

- La *curva* si trova in un sistema di coordinate non specificato e viene specificato il parametro *unità*.
- La *curva* si trova in un sistema di coordinate proiettate e viene specificata un'unità angolare.

ST_Length

- La *curva* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale geodetico, e viene specificata un'unità lineare.
- La *curva* si trova in un sistema di riferimento spaziale geodetico e viene specificata un'unità angolare.

Tipo restituito:

DOUBLE

Esempi:

Le seguenti istruzioni SQL consentono di creare una tabella SAMPLE_GEOMETRIES e di inserire una linea e una multilinea nella tabella.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_geometries(id SMALLINT, spatial_type varchar(20),  
    geometry ST_GEOMETRY)
```

```
INSERT INTO sample_geometries(id, spatial_type, geometry)  
VALUES  
    (1110, 'ST_LineString', ST_LineString('linestring(50 10, 50 20)', 1)),  
    (1111, 'ST_MultiLineString', ST_MultiLineString('multilinestring  
        ((33 2, 34 3, 35 6),  
        (28 4, 29 5, 31 8, 43 12),  
        (39 3, 37 4, 36 7))', 1))
```

Esempio 1:

La seguente istruzione SELECT consente di calcolare la lunghezza della linea nella tabella SAMPLE_GEOMTRIES.

```
SELECT id, spatial_type, cast(ST_Length(geometry..ST_ToLineString)  
    AS DECIMAL(7, 2)) AS "Line Length"  
FROM sample_geometries  
WHERE id = 1110
```

Risultati:

ID	SPATIAL_TYPE	Line Length
1110	ST_LineString	10.00

Esempio 2:

La seguente istruzione SELECT consente di calcolare la lunghezza della multilinea nella tabella SAMPLE_GEOMTRIES.

```
SELECT id, spatial_type, ST_Length(ST_ToMultiLine(geometry))  
    AS multiline_length  
FROM sample_geometries  
WHERE id = 1111
```

Risultati:

ID	SPATIAL_TYPE	MULTILINE_LENGTH
1111	ST_MultiLineString	+2.76437123387202E+001

ST_LineFromText

ST_LineFromText utilizza la rappresentazione WKT (well-known text) di una linea ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la linea corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La versione preferita per questa funzionalità è ST_LineString.

Sintassi:

```
db2gse.ST_LineFromText(wkt [, srs_id])
```

Parametro:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della linea ottenuta come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la linea ottenuta come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_LineString

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito utilizza la funzione ST_LineFromText per creare ed inserire una linea da una rappresentazione di linea WKT (well-known text). Le righe vengono inserite nella tabella SAMPLE_LINES con un ID e un valore di linea nel sistema di riferimento spaziale 1, nella rappresentazione WKT.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_lines(id SMALLINT, geometry ST_LineString)

INSERT INTO sample_lines(id, geometry)
VALUES
  (1110, ST_LineFromText('linestring(850 250, 850 850)', 1) ),
  (1111, ST_LineFromText('linestring empty', 1) )

SELECT id, cast(geometry..ST_AsText AS varchar(75)) AS linestring
FROM sample_lines
```

Risultati:

ST_LineFromText

```
ID      LINESTRING
-----
1110 LINESTRING ( 850.00000000 250.00000000, 850.00000000 850.00000000)
1111 LINESTRING EMPTY
```

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515

ST_LineFromWKB

ST_LineFromWKB utilizza la rappresentazione WKB (well-known binary) di una linea ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la linea corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La versione preferita per questa funzionalità è ST_LineString.

Sintassi:

```
►► db2gse.ST_LineFromWKB ( ( wkb ) [ , srs_id ] ) ►►
```

Parametro:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della multilinea ottenuta come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la linea ottenuta come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_LineString

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Il codice riportato di seguito utilizza la funzione ST_LineFromWKB per creare ed inserire una linea da una rappresentazione WKB (well-known binary). La riga viene inserita nella tabella SAMPLE_LINES con un ID e una linea nel sistema di riferimento spaziale 1 nella rappresentazione WKB.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_lines(id SMALLINT, geometry ST_LineString, wkb BLOB(32k))
```

```
INSERT INTO sample_lines(id, geometry)
```

```
VALUES
```

```
(1901, ST_LineString('linestring(850 250, 850 850)', 1) ),
```

```
(1902, ST_LineString('linestring(33 2, 34 3, 35 6)', 1) )

UPDATE sample_lines AS temp_correlated
SET   wkb = geometry.ST_AsBinary
WHERE id = temp_correlated.id

SELECT id, cast(ST_LineFromWKB(wkb)..ST_AsText AS varchar(90)) AS line
FROM   sample_lines
```

Risultati:

```
ID      LINE
-----
1901 LINESTRING ( 850.00000000 250.00000000, 850.00000000 850.00000000)

1902 LINESTRING ( 33.00000000 2.00000000, 34.00000000 3.00000000,
35.00000000 6.00000000)
```

Riferimenti correlati:

- “Rappresentazione WKB (well-known binary)” a pagina 520

ST_LineString

ST_LineString consente di creare una linea da uno dei seguenti input:

- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape ESRI
- Una rappresentazione in GML (geography markup language)

Per identificare il sistema di riferimento spaziale in cui verrà inserita la linea ottenuta come risultato, è possibile specificare un identificativo di sistema.

Se la rappresentazione WKT, WKB, GML o la rappresentazione shape ESRI sono nulle, viene restituito un valore nullo.

Sintassi:

```
db2gse.ST_LineString( ( wkt | wkb | shape | gml ) , -srs_id )
```

Parametro:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del poligono ottenuto come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del poligono ottenuto come risultato.
- shape* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape ESRI del poligono ottenuto come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta il poligono ottenuto come risultato utilizzando il linguaggio GML.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il poligono ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

ST_LineString

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene restituito un errore (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_LineString

Esempi:

Il codice riportato di seguito utilizza la funzione ST_LineString per creare ed inserire una linea da una rappresentazione di linea WKT (well-known text) o da una rappresentazione WKB (well-known binary).

L'esempio riportato di seguito consente di inserire una riga con ID nella tabella SAMPLE_LINES e una linea nel sistema di riferimento spaziale uno 1, nelle rappresentazioni WKT e GML.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

CREATE TABLE sample_lines(id SMALLINT, geometry ST_LineString)

INSERT INTO sample_lines(id, geometry)
VALUES
  (1110, ST_LineString('linestring(850 250, 850 850)', 1) ),
  (1111, ST_LineString('<gml:LineString srsName=";EPSG:4269";><gml:coord>
    <gml:X>90</gml:X><gml:Y>90</gml:Y>
    </gml:coord><gml:coord><gml:X>100</gml:X>
    <gml:Y>100</gml:Y></gml:coord>
    </gml:LineString>', 1) )

SELECT id, cast(geometry..ST_AsText AS varchar(75)) AS linestring
FROM sample_lines
```

Risultati:

```
ID      LINESTRING
-----
1110 LINESTRING ( 850.000000000 250.000000000, 850.000000000 850.000000000)
1111 LINESTRING ( 90.000000000 90.000000000, 100.000000000 100.000000000)
```

Riferimenti correlati:

- “Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati” a pagina 295
- “Rappresentazione WKT (well-known text)” a pagina 515

ST_LineStringN

ST_LineStringN utilizza una multilinea ed un indice come parametri di input e restituisce la multilinea identificata dall'indice. La linea ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della multilinea data.

Se la multilinea ha valore nullo o è vuota, o se l'indice è minore di 1 o maggiore del numero di linee, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►► db2gse.ST_LineStringN(*—multi_linea—*, *—indice—*) ◀◀

Parametro:

multi_linea

Un valore di tipo ST_MultiLineString che rappresenta la multilinea da cui viene restituito la linea identificata dall'*indice*.

indice Un valore di tipo INTEGER che identifica l'*ennesima* linea restituita da *multi_linea*.

Se l'*indice* è minore di 1 o maggiore del numero di linee in *multi_linea*, viene restituito un valore nullo e una condizione di avviso (SQLSTATE 01HS0).

Tipo restituito:

db2gse.ST_LineString

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'istruzione SELECT mostra come scegliere la seconda forma geometrica nella multilinea nella tabella SAMPLE_MLINES.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
```

```
CREATE TABLE sample_mlines (id INTEGER,
  geometry ST_MULTILINESTRING)
```

```
INSERT INTO sample_mlines(id, geometry)
VALUES
  (1110, ST_MultiLineString('multilineestring
    ((33 2, 34 3, 35 6),
     (28 4, 29 5, 31 8, 43 12),
     (39 3, 37 4, 36 7))', 1) ),
  (1111, ST_MLineFromText('multilineestring(
    (61 2, 64 3, 65 6),
    (58 4, 59 5, 61 8),
    (69 3, 67 4, 66 7, 68 9))', 1) )
```

```
SELECT id, cast(ST_LineStringN(geometry, 2)..ST_AsText
  AS varchar(110)) AS second_linestring
FROM   sample_mlines
```

Risultati:

ID	SECOND_LINestring
1110	LINESTRING (28.00000000 4.00000000, 29.00000000 5.00000000, 31.00000000 8.00000000, 43.00000000 12.00000000)
1111	LINESTRING (58.00000000 4.00000000, 59.00000000 5.00000000, 61.00000000 8.00000000)

Riferimenti correlati:

- "ST_NumLineStrings" a pagina 457

ST_M

ST_M è in grado di:

- Utilizzare un punto come parametro di input e restituirne la coordinata M (misura)
- Utilizzare un punto e una coordinata M e restituire il punto con la coordinata M impostata sull'unità di misura specificata, anche se il punto non dispone di coordinata M.

Se la coordinata M specificata è nulla, verrà rimossa dal punto.

Se il punto specificato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►► db2gse.ST_M (*punto* [, *coordinata_m*]) ►►

Parametri:

punto Un valore di tipo ST_Point del quale viene restituita o modificata la coordinata M.

coordinata_m

Un valore di tipo DOUBLE che rappresenta la nuova coordinata M per il *punto*.

Se la *coordinata_m* è nulla, la coordinata M verrà rimossa dal *punto*.

Tipi restituiti:

- DOUBLE, se la *coordinata_m* non viene specificata
- db2gse.ST_Point, se la *coordinata_m* viene specificata

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_M. Vengono creati tre punti e inseriti nella tabella SAMPLE_POINTS. Tutti appartengono al sistema di riferimento spaziale con ID 1.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points
VALUES (1, ST_Point (2, 3, 32, 5, 1))
```

```
INSERT INTO sample_points
VALUES (2, ST_Point (4, 5, 20, 4, 1))
```

```
INSERT INTO sample_points
VALUES (3, ST_Point (3, 8, 23, 7, 1))
```

Esempio 1:

Questo esempio consente di individuare la coordinata M dei punti contenuti nella tabella SAMPLE_POINTS.


```
SELECT id, ST_M (geometry) M_COORD
FROM sample_points
```

Risultati:

ID	M_COORD
1	+5.000000000000000E+000
2	+4.000000000000000E+000
3	+7.000000000000000E+000

Esempio 2:

Questo esempio restituisce uno dei punti aventi coordinata M impostata su 40.

```
SELECT id, CAST (ST_AsText (ST_M (geometry, 40) )
AS VARCHAR(60) ) M_COORD_40
FROM sample_points
WHERE id=3
```

Risultati:

ID	M_COORD_40
3	POINT ZM (3.00000000 8.00000000 23.00000000 40.00000000)

Riferimenti correlati:

- “ST_X” a pagina 503
- “ST_Y” a pagina 504
- “ST_Z” a pagina 506

ST_MaxM

ST_MaxM utilizza una forma geometrica come parametro di input e restituisce le relative coordinate M maggiori.

Se la forma geometrica data ha valore nullo o è vuota, o se non ha coordinate M, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_MaxM—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate M maggiori.

Tipo restituito:

DOUBLE

Esempi:

ST_MaxM

Gli esempi seguenti mostrano come utilizzare la funzione ST_MaxM. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                     110 140 22 3,
                                     120 130 26 4,
                                     110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                     0 4 35 9,
                                     5 4 32 12,
                                     5 0 31 5,
                                     0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                     8 4 10 12,
                                     9 4 12 11,
                                     12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata M maggiore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MaxM(geometry) AS INTEGER) MAX_M
FROM sample_polys
```

Risultati:

ID	MAX_M
1	4
2	12
3	16

Esempio 2:

Questo esempio consente di individuare la coordinata M maggiore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MAX ( ST_MaxM(geometry) ) AS INTEGER) OVERALL_MAX_M
FROM sample_polys
```

Risultati:

OVERALL_MAX_M
16

Argomenti correlati:

- “ST_MaxX” a pagina 425

Riferimenti correlati:

- “ST_MaxY” a pagina 426
- “ST_MaxZ” a pagina 428
- “ST_MinM” a pagina 435

ST_MaxX

ST_MaxX utilizza una forma geometrica come parametro di input e restituisce le relative coordinate X maggiori.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_MaxX—(*—forma geometrica—*)—►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate X maggiori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MaxX. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS. Il terzo esempio mostra come utilizzare tutte le funzioni che restituiscono i valori delle coordinate minima e massima per valutare l'intervallo spaziale delle forme geometriche memorizzate in una determinata colonna spaziale.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                8 4 10 12,
                                9 4 12 11,
                                12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata X maggiore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MaxX(geometry) AS INTEGER) MAX_X_COORD
FROM sample_polys
```

ST_MaxX

Risultati:

ID	MAX_X_COORD
1	120
2	5
3	12

Esempio 2:

Questo esempio consente di individuare la coordinata X maggiore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MAX ( ST_MaxX(geometry) ) AS INTEGER) OVERALL_MAX_X
FROM sample_polys
```

Risultati:

OVERALL_MAX_X
120

Esempio 3:

Questo esempio consente di individuare l'estensione spaziale (dal totale minimo al totale massimo) di tutti i poligoni nella tabella SAMPLE_POLYS. Questo calcolo viene utilizzato generalmente per paragonare l'estensione spaziale reale delle forme geometriche all'estensione del sistema di riferimento spaziale associato ai dati, in modo da determinare se esiste spazio affinché i dati possano svilupparsi.

```
SELECT CAST ( MIN ( ST_MinX (geometry)) AS INTEGER) MIN_X,
CAST ( MIN ( ST_MinY (geometry)) AS INTEGER) MIN_Y,
CAST ( MIN ( ST_MinZ (geometry)) AS INTEGER) MIN_Z,
CAST ( MIN ( ST_MinM (geometry)) AS INTEGER) MIN_M,
CAST ( MAX ( ST_MaxX (geometry)) AS INTEGER) MAX_X,
CAST ( MAX ( ST_MaxY (geometry)) AS INTEGER) MAX_Y,
CAST ( MAX ( ST_MaxZ (geometry)) AS INTEGER) MAX_Z,
CAST ( MAX ( ST_MaxM(geometry)) AS INTEGER) MAX_M,
FROM sample_polys
```

Risultati:

MIN_X	MIN_Y	MIN_Z	MIN_M	MAX_X	MAX_Y	MAX_Z	MAX_M
0	0	10	3	120	140	40	16

Riferimenti correlati:

- "ST_MaxM" a pagina 423
- "ST_MaxY" a pagina 426
- "ST_MaxZ" a pagina 428
- "ST_MinX" a pagina 436

ST_MaxY

ST_MaxY utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Y maggiori.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►► db2gse.ST_MaxY(*—forma geometrica—*)►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate Y maggiori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MaxY. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                8 4 10 12,
                                9 4 12 11,
                                12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Y maggiore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MaxY(geometry) AS INTEGER) MAX_Y
FROM sample_polys
```

Risultati:

ID	MAX_Y
1	140
2	4
3	13

Esempio 2:

Questo esempio consente di individuare la coordinata Y maggiore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MAX ( ST_MaxY(geometry) ) AS INTEGER) OVERALL_MAX_Y
FROM sample_polys
```

Risultati:

ST_MaxY

OVERALL_MAX_Y

140

Argomenti correlati:

- “ST_MaxX” a pagina 425

Riferimenti correlati:

- “ST_MaxM” a pagina 423
- “ST_MaxZ” a pagina 428
- “ST_MinY” a pagina 438

ST_MaxZ

ST_MaxZ utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Z maggiori.

Se la forma geometrica data ha valore nullo o è vuota, o se non ha coordinate Z, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_MaxZ—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate Z maggiori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MaxZ. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
```

```
8 4 10 12,
9 4 12 11,
12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Z maggiore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MaxZ(geometry) AS INTEGER) MAX_Z
FROM sample_polys
```

Risultati:

ID	MAX_Z
1	26
2	40
3	12

Esempio 2:

Questo esempio consente di individuare la coordinata Z maggiore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MAX ( ST_MaxZ(geometry) ) AS INTEGER) OVERALL_MAX_Z
FROM sample_polys
```

Risultati:

OVERALL_MAX_Z
40

Argomenti correlati:

- “ST_MaxX” a pagina 425

Riferimenti correlati:

- “ST_MaxM” a pagina 423
- “ST_MaxY” a pagina 426
- “ST_MinZ” a pagina 439

ST_MBR

ST_MBR utilizza una forma geometrica come parametro di input e ne restituisce il rettangolo circoscritto minore.

Se la forma geometrica data è un punto, viene restituito lo stesso punto. Se la forma geometrica è una linea orizzontale o verticale e il sistema di riferimento spaziale non è geodetico, verrà restituita la stessa linea verticale o orizzontale. Diversamente, viene restituito il rettangolo minimo circoscritto alla forma geometrica come poligono. Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_MBR—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui viene restituito l'MBR (minimum bounding rectangle).

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Questo esempio mostra come utilizzare la funzione ST_MBR per restituire l'MBR (minimum bounding rectangle) di un poligono. Poiché la forma geometrica specificata è un poligono, l'MBR viene restituito come poligono.

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

INSERT INTO sample_polys
VALUES (1, ST_Polygon ('polygon (( 5 5, 7 7, 5 9, 7 9, 9 11, 13 9,
                               15 9, 13 7, 15 5, 9 6, 5 5))', 0) )

INSERT INTO sample_polys
VALUES (2, ST_Polygon ('polygon (( 20 30, 25 35, 30 30, 20 30))', 0) )

SELECT id, CAST (ST_AsText ( ST_MBR(geometry)) AS VARCHAR(150) ) MBR
FROM sample_polys
```

Risultati:

ID	MBR
1	POLYGON ((5.00000000 5.00000000, 15.00000000 5.00000000, 15.00000000 11.00000000, 5.00000000 11.00000000, 5.00000000 5.00000000))
2	POLYGON ((20.00000000 30.00000000, 30.00000000 30.00000000, 30.00000000 35.00000000, 20.00000000 35.00000000, 20.00000000 30.00000000))

Riferimenti correlati:

- “ST_Envelope” a pagina 378
- “ST_MBRIntersects” a pagina 430

ST_MBRIntersects

ST_MBRIntersects utilizza due forme geometriche come parametri di input e se gli MBR delle due forme si intersecano, restituisce 1. In caso contrario, restituisce 0 (zero). L'MBR di un punto e di una linea verticale o orizzontale è rappresentato dalla forma geometrica stessa.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se una delle forme geometriche date è nulla o vuota, viene restituito un valore nullo.

Sintassi:

```
►►—db2gse.ST_MBRIntersects—(—forma geometrica 1—,—forma geometrica 2—)——►►
```

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'MBR si interseca con l'MBR della *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina se l'MBR si interseca con l'MBR della *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempi:

Questi esempi mostrano come utilizzare la funzione ST_MBRIntersects per ottenere un'approssimazione della vicinanza di due poligoni non intersecati tra loro, calcolata sulla base dell'intersezione dei rispettivi MBR. Il primo esempio utilizza l'espressione SQL CASE. Il secondo esempio utilizza una singola istruzione SELECT per rilevare quei poligono che intersecano l'MBR del poligono con ID = 2.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon ('polygon (( 0 0, 30 0, 40 30, 40 35,
                               5 35, 5 10, 20 10, 20 5, 0 0 ))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon ('polygon (( 15 15, 15 20, 60 20, 60 15,
                               15 15 ))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon ('polygon (( 115 15, 115 20, 160 20, 160 15,
                               115 15 ))', 0) )
```

Esempio 1:

La seguente istruzione SELECT utilizza l'espressione CASE per individuare gli ID dei poligoni i cui MBR si intersecano.

```
SELECT a.id, b.id,
       CASE ST_MBRIntersects (a.geometry, b.geometry)
         WHEN 0 THEN 'MBRs do not intersect'
         WHEN 1 THEN 'MBRs intersect'
       END AS MBR_INTERSECTS
FROM   sample_polys a, sample_polys b
WHERE  a.id <= b.id
```

Risultati:

ST_MBRIntersects

ID	ID	MBR_INTERSECTS
	1	1 MBRs intersect
	1	2 MBRs intersect
	2	2 MBRs intersect
	1	3 MBRs do not intersect
	2	3 MBRs do not intersect
	3	3 MBRs intersect

Esempio 2:

La seguente istruzione SELECT consente di stabilire se gli MBR delle forme geometriche si intersecano con gli MBR del poligono con ID = 2.

```
SELECT a.id, b.id, ST_MBRIntersects (a.geometry, b.geometry) MBR_INTERSECTS
FROM sample_polys a, sample_polys b
WHERE a.id = 2
```

Risultati

ID	ID	MBR_INTERSECTS
	2	1
	2	2
	2	3

Riferimenti correlati:

- "ST_EnvIntersects" a pagina 379
- "ST_MBR" a pagina 429

ST_MeasureBetween, ST_LocateBetween

ST_MeasureBetween o ST_LocateBetween utilizzano una forma geometrica e due coordinate M (misure) come parametri di input e restituiscono la parte della forma geometrica data che rappresenta la serie di percorsi o punti disconnessi tra le due coordinate M.

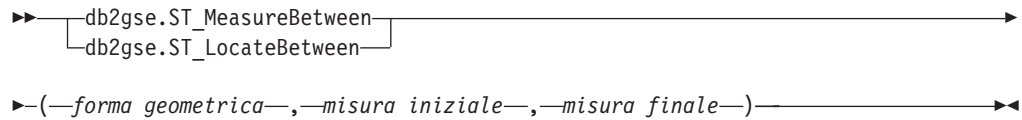
Per le curve, le multicurve, le superfici e le multisuperfici, per ottenere il risultato viene eseguita una interpolazione. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se la forma geometrica data è rappresentata da una superficie o una multisuperficie, ST_MeasureBetween o ST_LocateBetween verranno applicate alla circonferenza interna ed esterna della forma geometrica. Se nessuna parte della forma geometrica data si trova nell'intervallo definito dalle coordinate M, viene restituita una forma geometrica vuota. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La forma geometrica ottenuta come risultato viene rappresentata nel tipo spaziale più adatto. Se possibile, verrà utilizzato un punto, una linea o un poligono, in caso contrario, verrà utilizzato un multipunto, una multilinea o un multipoligono.

Le due funzioni possono essere richiamate come metodi.

Sintassi:

**Parametri:***forma geometrica*

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica in cui individuare le misure comprese tra *misura iniziale* e *misura finale*.

misura iniziale

Un valore di tipo DOUBLE che rappresenta il limite inferiore dell'intervallo di misure. Se il valore è nullo, non viene applicato alcun limite inferiore.

misura finale

Un valore di tipo DOUBLE che rappresenta il limite superiore dell'intervallo di misure. Se il valore è nullo, non viene applicato alcun limite superiore.

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

La coordinata M (misura) delle forme geometriche, viene definita dall'utente. E' una coordinata variabile che può rappresentare qualsiasi elemento che si desidera misurare; ad esempio, una distanza, la temperatura, la pressione o valori di pH.

Questo esempio mostra come utilizzare le coordinate M per registrare i dati raccolti dalle misurazione di valori di pH. Il pH del suolo viene rilevato lungo un'autostrada in determinati punti. In base alle procedure standard, in ogni punto in cui viene effettuato un rilevamento, si prende nota delle coordinate X e Y del punto e il valore del pH.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, geometry ST_LineString)
```

```
INSERT INTO sample_lines
VALUES (1, ST_LineString ('linestring m (2 2 3, 3 5 3,
                          3 3 6, 4 4 6,
                          5 5 6, 6 6 8)', 1 ) )
```

Per individuare il percorso in cui il livello di acidità del suolo varia da 4 a 6, utilizzare la seguente istruzione SELECT:

```
SELECT id, CAST( ST_AsText( ST_MeasureBetween( 4, 6 )
AS VARCHAR(150) ) MEAS_BETWEEN_4_AND_6
FROM sample_lines
```

Risultati:

```
ID          MEAS_BETWEEN_4_AND_6
-----
1 LINESTRING M (3.00000000 4.33333300 4.00000000,
```

ST_MeasureBetween e ST_LocateBetween

```
3.00000000 3.00000000 6.00000000,  
4.00000000 4.00000000 6.00000000,  
5.00000000 5.00000000 6.00000000)
```

ST_MidPoint

ST_MidPoint utilizza una curva come parametro di input e restituisce il punto della curva equidistante dai punti esterni. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della curva data.

Se la curva è vuota, viene restituito un punto vuoto. Se la curva data è nulla, viene restituito un valore nullo.

Se la curva contiene coordinate Z o M (misure), il punto centrale viene determinato solo dai valori delle coordinate X e Y della curva. La coordinata Z e la misura del punto restituito verranno interpolate.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_MidPoint—(—curva—)—————►►
```

Parametro:

curva Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la curva per la quale viene restituito il punto centrale.

Tipo restituito:

db2gse.ST_Point

Esempio:

Questo esempio mostra come utilizzare la funzione ST_MidPoint per ottenere il punto centrale delle curve.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse  
CREATE TABLE sample_lines (id INTEGER, geometry ST_LineString)  
  
INSERT INTO sample_lines (id, geometry)  
VALUES (1, ST_LineString ('linestring (0 0, 0 10, 0 20, 0 30, 0 40)', 1 ) )  
  
INSERT INTO sample_lines (id, geometry)  
VALUES (2, ST_LineString ('linestring (2 2, 3 5, 3 3, 4 4, 5 5, 6 6)', 1 ) )  
  
INSERT INTO sample_lines (id, geometry)  
VALUES (3, ST_LineString ('linestring (0 10, 0 0, 10 0, 10 10)', 1 ) )  
  
INSERT INTO sample_lines (id, geometry)  
VALUES (4, ST_LineString ('linestring (0 20, 5 20, 10 20, 15 20)', 1 ) )  
  
SELECT id, CAST( ST_AsText( ST_MidPoint(geometry) ) AS VARCHAR(60) ) MID_POINT  
FROM sample_lines
```

Risultati:

```
ID          MID_POINT  
-----  
1 POINT ( 0.00000000 20.00000000)
```

```

2 POINT ( 3.00000000 3.45981800)
3 POINT ( 5.00000000 0.00000000)
4 POINT ( 7.50000000 20.00000000)

```

ST_MinM

ST_MinM utilizza una forma geometrica come parametro di input e restituisce le relative coordinate M minori.

Se la forma geometrica data ha valore nullo o è vuota, o se non ha coordinate M, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```

▶▶—db2gse.ST_MinM—(—forma geometrica—)—————▶▶

```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate M minori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MinM. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```

SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

```

```

INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )

```

```

INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )

```

```

INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                8 4 10 12,
                                9 4 12 11,
                                12 13 10 16))', 0) )

```

Esempio 1:

Questo esempio consente di individuare la coordinata M minore di ciascun poligono contenuto in SAMPLE_POLYS.

ST_MinM

```
SELECT id, CAST ( ST_MinM(geometry) AS INTEGER) MIN_M
FROM sample_polys
```

Risultati:

ID	MIN_M
1	3
2	5
3	11

Esempio 2:

Questo esempio consente di individuare la coordinata M minore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MIN ( ST_MinM(geometry) ) AS INTEGER) OVERALL_MIN_M
FROM sample_polys
```

Risultati:

OVERALL_MIN_M
3

Riferimenti correlati:

- "ST_MaxM" a pagina 423
- "ST_MinX" a pagina 436
- "ST_MinY" a pagina 438
- "ST_MinZ" a pagina 439

ST_MinX

ST_MinX utilizza una forma geometrica come parametro di input e restituisce le relative coordinate X minori.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_MinX—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate X minori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MinX. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                8 4 10 12,
                                9 4 12 11,
                                12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata X minore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MinX(geometry) AS INTEGER) MIN_X
FROM sample_polys
```

Risultati:

ID	MIN_X
1	110
2	0
3	8

Esempio 2:

Questo esempio consente di individuare la coordinata X minore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MIN ( ST_MinX(geometry) ) AS INTEGER) OVERALL_MIN_X
FROM sample_polys
```

Risultati:

OVERALL_MIN_X
0

Argomenti correlati:

- “ST_MaxX” a pagina 425

Riferimenti correlati:

- “ST_MinM” a pagina 435
- “ST_MinY” a pagina 438
- “ST_MinZ” a pagina 439

ST_MinY

ST_MinY utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Y minori.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_MinY—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate Y minori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MinY. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                                110 140 22 3,
                                120 130 26 4,
                                110 120 20 3))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                                0 4 35 9,
                                5 4 32 12,
                                5 0 31 5,
                                0 0 40 7))', 0) )
```

```
INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                                8 4 10 12,
                                9 4 12 11,
                                12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Y minore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MinY(geometry) AS INTEGER) MIN_Y
FROM sample_polys
```

Risultati:

ID	MIN_Y
1	120
2	0
3	4

Esempio 2:

Questo esempio consente di individuare la coordinata Y minore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MIN ( ST_MinY(geometry) ) AS INTEGER) OVERALL_MIN_Y
FROM sample_polys
```

Risultati:

```
OVERALL_MIN_Y
-----
0
```

Riferimenti correlati:

- “ST_MaxY” a pagina 426
- “ST_MinM” a pagina 435
- “ST_MinX” a pagina 436
- “ST_MinZ” a pagina 439

ST_MinZ

ST_MinZ utilizza una forma geometrica come parametro di input e restituisce le relative coordinate Z minori.

Se la forma geometrica data ha valore nullo o è vuota, o se non ha coordinate Z, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_MinZ—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari di cui vengono restituite le coordinate Z minori.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_MinZ. Vengono creati tre poligoni e inseriti nella tabella SAMPLE_POLYS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
```

ST_MinZ

```
VALUES (1, ST_Polygon('polygon zm ((110 120 20 3,
                               110 140 22 3,
                               120 130 26 4,
                               110 120 20 3))', 0) )

INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon zm ((0 0 40 7,
                               0 4 35 9,
                               5 4 32 12,
                               5 0 31 5,
                               0 0 40 7))', 0) )

INSERT INTO sample_polys
VALUES (3, ST_Polygon('polygon zm ((12 13 10 16,
                               8 4 10 12,
                               9 4 12 11,
                               12 13 10 16))', 0) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Z minore di ciascun poligono contenuto in SAMPLE_POLYS.

```
SELECT id, CAST ( ST_MinZ(geometry) AS INTEGER) MIN_Z
FROM sample_polys
```

Risultati:

ID	MIN_Z
1	20
2	31
3	10

Esempio 2:

Questo esempio consente di individuare la coordinata Z minore di tutti i poligoni contenuti nella colonna GEOMETRY.

```
SELECT CAST ( MIN ( ST_MinZ(geometry) ) AS INTEGER) OVERALL_MIN_Z
FROM sample_polys
```

Risultati:

OVERALL_MIN_Z
10

Riferimenti correlati:

- “ST_MaxZ” a pagina 428
- “ST_MinM” a pagina 435
- “ST_MinX” a pagina 436
- “ST_MinY” a pagina 438

ST_MLineFromText

ST_MLineFromText utilizza la rappresentazione WKT (well-known text) di una multilinea ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la multilinea corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiLineString. La flessibilità di ST_MultiLineString consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
db2gse.ST_MLineFromText( ( wkt [, srs_id ] ) )
```

Parametri:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della multilinea ottenuta come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la multilinea ottenuta come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiLineString

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MLineFromText per creare ed inserire una multilinea dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà una multilinea nel sistema di riferimento spaziale 1. La multilinea appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono:

- Linea 1: (33, 2) (34, 3) (35, 6)
- Linea 2: (28, 4) (29, 5) (31, 8) (43, 12)
- Linea 3: (39, 3) (37, 4) (36, 7)

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mlines (id INTEGER, \geometry ST_MultiLineString)
```

```
INSERT INTO sample_mlines
VALUES (1110, ST_MLineFromText ('multilineestring ( (33 2, 34 3, 35 6),
(28 4, 29 5, 31 8, 43 12),
(39 3, 37 4, 36 7) )', 1) )
```

La seguente istruzione SELECT restituisce la multilinea registrata nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(280) ) MULTI_LINE_STRING
FROM sample_mlines
WHERE id = 1110
```

Risultati:

ST_MLineFromText

```
ID          MULTI_LINE_STRING
-----
1110 MULTILINESTRING (( 33.00000000 2.00000000, 34.00000000 3.00000000,
                        35.00000000 6.00000000),
                        ( 28.00000000 4.00000000, 29.00000000 5.00000000,
                        31.00000000 8.00000000, 43.00000000 12.00000000),
                        ( 39.00000000 3.00000000, 37.00000000 4.00000000,
                        36.00000000 7.00000000 ))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MLineFromWKB” a pagina 442
- “ST_MultiLineString” a pagina 450

ST_MLineFromWKB

ST_MLineFromWKB utilizza la rappresentazione WKB (well-known binary) di una multilinea ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce la multilinea corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiLineString. La flessibilità di ST_MultiLineString consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
►► db2gse.ST_MLineFromWKB ( ( wkb [ , srs_id ] ) )
```

Parametri:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della multilinea ottenuta come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la multilinea ottenuta come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiLineString

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MLineFromWKB per creare una multilinea dalla corrispondente rappresentazione WKB. La forma geometrica è una multilinea del sistema di riferimento spaziale 1. In questo esempio, la multilinea viene memorizzata con ID = 10 nella colonna GEOMETRY della tabella SAMPLE_MLINES, quindi la colonna WKB viene aggiornata col la rispettiva rappresentazione WKB (utilizzando la funzione ST_AsBinary). Infine, per restituire la multilinea dalla colonna WKB, viene utilizzata la funzione ST_MLineFromWKB. Le coordinate X e Y della forma geometrica sono:

- Linea 1: (61, 2) (64, 3) (65, 6)
- Linea 2: (58, 4) (59, 5) (61, 8)
- Linea 3: (69, 3) (67, 4) (66, 7) (68, 9)

La tabella SAMPLE_MLINES contiene la colonna GEOMETRY, in cui è memorizzata la multilinea, e la colonna WKB, in cui è memorizzata la rappresentazione WKB della multilinea.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mlines (id INTEGER, geometry ST_MultiLineString,
wkb BLOB(32K))
```

```
INSERT INTO sample_mlines
VALUES (10, ST_MultiLineString ('multilinestring
( (61 2, 64 3, 65 6),
(58 4, 59 5, 61 8),
(69 3, 67 4, 66 7, 68 9) )', 1) )
```

```
UPDATE sample_mlines AS temporary_correlated
SET wkb = ST_AsBinary( geometry )
WHERE id = temporary_correlated.id
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_MLineFromWKB per richiamare la multilinea dalla colonna WKB.

```
SELECT id, CAST( ST_AsText( ST_MLineFromWKB (wkb) )
AS VARCHAR(280) ) MULTI_LINE_STRING
FROM sample_mlines
WHERE id = 10
```

Risultati:

```
ID          MULTI_LINE_STRING
-----
10 MULTILINESTRING (( 61.00000000 2.00000000, 64.00000000 3.00000000,
65.00000000 6.00000000),
( 58.00000000 4.00000000, 59.00000000 5.00000000,
61.00000000 8.00000000),
( 69.00000000 3.00000000, 67.00000000 4.00000000,
66.00000000 7.00000000, 68.00000000 9.00000000 ))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MLineFromText” a pagina 440
- “ST_MultiLineString” a pagina 450
- “Rappresentazione WKB (well-known binary)” a pagina 520

ST_MPointFromText

ST_MPointFromText utilizza la rappresentazione WKT (well-known text) di un multipunto ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il multipunto corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiPoint. La flessibilità di ST_MultiPoint consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
db2gse.ST_MPointFromText( ( wkt [, srs_id ] ) )
```

Parametri:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del multipunto ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipunto ottenuto come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiPoint

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MPointFromText per creare ed inserire un multipunto dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un multipunto nel sistema di riferimento spaziale 1. Il multipunto appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono: (1, 2) (4, 3) (5, 6).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpoints (id INTEGER, geometry ST_MultiPoint)
```

```
INSERT INTO sample_mpoints
VALUES (1110, ST_MPointFromText ('multipoint (1 2, 4 3, 5 6)'), 1)
```

La seguente istruzione SELECT restituisce il multipunto registrato nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(280) ) MULTIPOINT
FROM sample_mpoints
WHERE id = 1110
```

Risultati:

```
ID          MULTIPOINT
-----
1110 MULTIPOINT (1.00000000 2.00000000, 4.00000000 3.00000000,
                5.00000000 6.00000000)
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MPointFromWKB” a pagina 445
- “ST_MultiPoint” a pagina 451
- “Rappresentazione WKT (well-known text)” a pagina 515

ST_MPointFromWKB

ST_MPointFromWKB utilizza la rappresentazione WKB (well-known binary) di un multipunto ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il multipunto corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiPoint. La flessibilità di ST_MultiPoint consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
►► db2gse.ST_MPointFromWKB ( ( wkb [ , srs_id ] ) ) ◀◀
```

Parametri:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del multipunto ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipunto ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiPoint

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

ST_MPointFromWKB

Questo esempio mostra come utilizzare ST_MPointFromWKB per creare un multipunto dalla corrispondente rappresentazione WKB. La forma geometrica è un multipunto del sistema di riferimento spaziale 1. In questo esempio, il multipunto viene memorizzato con ID = 10 nella colonna GEOMETRY della tabella SAMPLE_MPOINTS, quindi la colonna WKB viene aggiornata col la rispettiva rappresentazione WKB (utilizzando la funzione ST_AsBinary). Infine, per restituire il multipunto dalla colonna WKB, viene utilizzata la funzione ST_MPointFromWKB. Le coordinate X e Y della forma geometrica sono: (44, 14) (35, 16) (24, 13).

La tabella SAMPLE_MPOINTS contiene la colonna GEOMETRY, in cui è memorizzato il multipunto, e la colonna WKB, in cui è memorizzata la rappresentazione WKB del multipunto.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpoints (id INTEGER, geometry ST_MultiPoint,
    wkb BLOB(32K))

INSERT INTO sample_mpoints
    VALUES (10, ST_MultiPoint ('multipoint ( 4 14, 35 16, 24 13)', 1))

UPDATE sample_mpoints AS temporary_correlated
    SET wkb = ST_AsBinary( geometry )
    WHERE id = temporary_correlated.id
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_MPointFromWKB per richiamare il multipunto dalla colonna WKB.

```
SELECT id, CAST( ST_AsText( ST_MLineFromWKB (wkb)) AS VARCHAR(100)) MULTIPOINT
    FROM sample_mpoints
    WHERE id = 10
```

Risultati:

ID	MULTIPOINT
10	MULTIPOINT (44.00000000 14.00000000, 35.00000000 16.00000000 24.00000000 13.00000000)

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MPointFromText” a pagina 444
- “ST_MultiPoint” a pagina 451
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “ST_Point” a pagina 465

ST_MPolyFromText

ST_MPolyFromText utilizza la rappresentazione WKT (well-known text) di un multipoligono ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il multipoligono corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiPolygon. La flessibilità di ST_MultiPolygon consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
►► db2gse.ST_MPolyFromText ( ( wkt , srs_id ) )
```

Parametri:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del multipoligono ottenuto come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipoligono ottenuto come risultato.
- Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).
- Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiPolygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MPolyFromText per creare ed inserire un multipoligono dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un multipoligono nel sistema di riferimento spaziale 1. Il multipoligono appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono:

- Poligono 1: (3, 3) (4, 6) (5, 3) (3, 3)
- Poligono 2: (8, 24) (9, 25) (1, 28) (8, 24)
- Poligono 3: (13, 33) (7, 36) (1, 40) (10, 43) (13, 33)

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpolys (id INTEGER, geometry ST_MultiPolygon)
```

```
INSERT INTO sample_mpolys
VALUES (1110,
       ST_MPolyFromText ('multipolygon (( (3 3, 4 6, 5 3, 3 3),
                                         (8 24, 9 25, 1 28, 8 24),
                                         (13 33, 7 36, 1 40, 10 43 13 33) ))', 1) )
```

La seguente istruzione SELECT restituisce il multipoligono registrato nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(350) ) MULTI_POLYGON
FROM sample_mpolys
WHERE id = 1110
```

Risultati:

ST_MPolyFromText

```
ID          MULTI_POLYGON
-----
1110 MULTIPOLYGON ((( 13.00000000 33.00000000, 10.00000000 43.00000000,
1.00000000 40.00000000, 7.00000000 36.00000000,
13.00000000 33.00000000)),
(( 8.00000000 24.00000000, 9.00000000 25.00000000,
1.00000000 28.00000000, 8.00000000 24.00000000)),
( 3.00000000 3.00000000, 5.00000000 3.00000000,
4.00000000 6.00000000, 3.00000000 3.00000000)))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MPolyFromWKB” a pagina 448
- “ST_MultiPolygon” a pagina 453
- “Rappresentazione WKT (well-known text)” a pagina 515

ST_MPolyFromWKB

ST_MPolyFromWKB utilizza la rappresentazione WKB (well-known binary) di un multipoligono ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il multipoligono corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_MultiPolygon. La flessibilità di ST_MultiPolygon consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
►►—db2gse.ST_MPolyFromWKB—(—wkb—└┬┘, —srs_id—)◄◄
```

Parametri:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del multipoligono ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipoligono ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* specificato non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiPolygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MPolyFromWKB per creare un multipoligono dalla corrispondente rappresentazione WKB. La forma geometrica è un multipoligono del sistema di riferimento spaziale 1. In questo esempio, il multipoligono viene memorizzato con ID = 10 nella colonna GEOMETRY della tabella SAMPLE_MPOLYS, quindi la colonna WKB viene aggiornata col la rispettiva rappresentazione WKB (utilizzando la funzione ST_AsBinary). Infine, per restituire il multipoligono dalla colonna WKB, viene utilizzata la funzione ST_MPolyFromWKB. Le coordinate X e Y della forma geometrica sono:

- Poligono 1: (1, 72) (4, 79) (5, 76) (1, 72)
- Poligono 2: (10, 20) (10, 40) (30, 41) (10, 20)
- Poligono 3: (9, 43) (7, 44) (6, 47) (9, 43)

La tabella SAMPLE_MPOLYS contiene la colonna GEOMETRY, in cui è memorizzato il multipoligono, e la colonna WKB, in cui è memorizzata la rappresentazione WKB del multipoligono.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpolys (id INTEGER,
    geometry ST_MultiPolygon, wkb BLOB(32K))
```

```
INSERT INTO sample_mpolys
VALUES (10, ST_MultiPolygon ('multipolygon
    (( (1 72, 4 79, 5 76, 1 72),
    (10 20, 10 40, 30 41, 10 20),
    (9 43, 7 44, 6 47, 9 43) ))', 1))
```

```
UPDATE sample_mpolys AS temporary_correlated
SET wkb = ST_AsBinary( geometry )
WHERE id = temporary_correlated.id
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_MPolyFromWKB per richiamare il multipoligono dalla colonna WKB.

```
SELECT id, CAST( ST_AsText( ST_MPolyFromWKB (wkb) )
    AS VARCHAR(320) ) MULTIPOLYGON
FROM sample_mpolys
WHERE id = 10
```

Risultati:

```
ID          MULTIPOLYGON
-----
10 MULTIPOLYGON ((( 10.00000000 20.00000000, 30.00000000
    41.00000000, 10.00000000 40.00000000, 10.00000000
    20.00000000)),
    ( 1.00000000 72.00000000, 5.00000000
    76.00000000, 4.00000000 79.00000000, 1.00000000
    72.00000000)),
    ( 9.00000000 43.00000000, 6.00000000
    47.00000000, 7.00000000 44.00000000, 9.00000000
    43.00000000 )))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_MPolyFromText” a pagina 446
- “ST_MultiPolygon” a pagina 453

ST_MPolyFromWKB

- “Rappresentazione WKB (well-known binary)” a pagina 520
- “ST_Polygon” a pagina 475

ST_MultiLineString

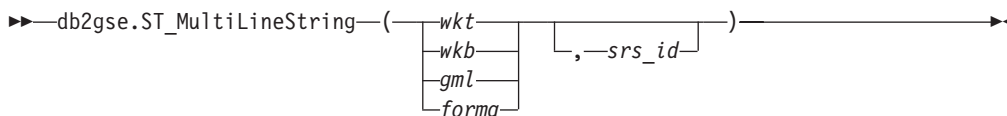
ST_MultiLineString consente di creare una multilinea da uno dei seguenti input:

- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape
- Una rappresentazione in GML (geography markup language)

Per identificare il sistema di riferimento spaziale in cui verrà inserita la linea ottenuta come risultato, è possibile specificare un identificativo di sistema.

Se la rappresentazione WKT, WKB, GML o la rappresentazione shape sono nulle, viene restituito un valore nullo.

Sintassi:



Parametri:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della multilinea ottenuta come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della multilinea ottenuta come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta la multilinea ottenuta come risultato utilizzando il linguaggio GML.
- forma* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape della multilinea ottenuta come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la multilinea ottenuta come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiLineString

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MultiLineString per creare ed inserire una multilinea dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà una multilinea nel sistema di riferimento spaziale 1. La multilinea appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono:

- Linea 1: (33, 2) (34, 3) (35, 6)
- Linea 2: (28, 4) (29, 5) (31, 8) (43, 12)
- Linea 3: (39, 3) (37, 4) (36, 7)

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mlines (id INTEGER,
                             geometry ST_MultiLineString)
```

```
INSERT INTO sample_mlines
VALUES (1110,
       ST_MultiLineString ('multilinestring ( (33 2, 34 3, 35 6),
                                             (28 4, 29 5, 31 8, 43 12),
                                             (39 3, 37 4, 36 7) )', 1) )
```

La seguente istruzione SELECT restituisce la multilinea registrata nella tabella:

```
SELECT id,
       CAST( ST_AsText( geometry ) AS VARCHAR(280) )
MULTI_LINE_STRING
FROM sample_mlines
WHERE id = 1110
```

Risultati:

```
ID          MULTI_LINE_STRING
-----
1110 MULTILINESTRING (( 33.00000000 2.00000000, 34.00000000 3.00000000,
                       35.00000000 6.00000000),
                       ( 28.00000000 4.00000000, 29.00000000 5.00000000,
                       31.00000000 8.00000000, 43.00000000 12.00000000),
                       ( 39.00000000 3.00000000, 37.00000000 4.00000000,
                       36.00000000 7.00000000 ))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_MultiPoint

ST_MultiPoint consente di creare un multipunto da uno dei seguenti input:

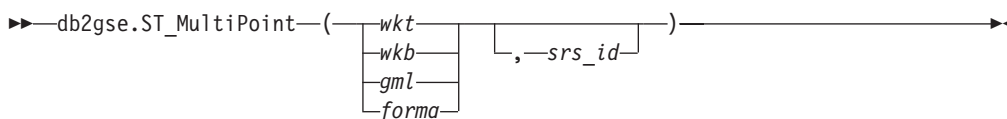
- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape
- Una rappresentazione in GML (geography markup language)

Per indicare il sistema di riferimento spaziale in cui verrà inserito il multipunto ottenuto come risultato, è possibile specificare un identificativo di sistema.

ST_MultiPoint

Se la rappresentazione WKT, WKB, GML o la rappresentazione shape sono nulle, viene restituito un valore nullo.

Sintassi:



Parametri:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del multipunto ottenuto come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del multipunto ottenuto come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta il multipunto ottenuto come risultato utilizzando il linguaggio GML.
- forma* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape del multipunto ottenuto come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipunto ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Point

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MultiPoint per creare ed inserire un multipunto dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un multipunto nel sistema di riferimento spaziale 1. Il multipunto appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono: (1, 2) (4, 3) (5, 6).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpoints (id INTEGER, geometry ST_MultiPoint)
```

```
INSERT INTO sample_mpoints
VALUES (1110, ST_MultiPoint ('multipoint (1 2, 4 3, 5 6) '), 1))
```

La seguente istruzione SELECT restituisce il multipunto registrato nella tabella:

```
SELECT id, CAST( ST_AsText(geometry) AS VARCHAR(90)) MULTIPOINT
FROM sample_mpoints
WHERE id = 1110
```

Risultati:

```
ID          MULTIPOINT
-----
1110 MULTIPOINT (1.00000000 2.00000000, 4.00000000
                3.00000000, 5.00000000 6.00000000)
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_MultiPolygon

ST_MultiPolygon consente di creare un multipoligono da uno dei seguenti input:

- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape
- Una rappresentazione in GML (geography markup language)

Per identificare il sistema di riferimento spaziale in cui verrà inserita il multipoligono ottenuto come risultato, è possibile specificare un identificativo di sistema.

Se la rappresentazione WKT, WKB, GML o la rappresentazione shape sono nulle, viene restituito un valore nullo.

Sintassi:

```
►► db2gse.ST_MultiPolygon(wkt,  
                        [wkb],  
                        [shape],  
                        [gml],  
                        [, srs_id])
```

Parametri:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del multipoligono ottenuto come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del multipoligono ottenuto come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta il multipoligono ottenuto come risultato utilizzando il linguaggio GML.
- shape* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape del multipoligono ottenuto come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il multipoligono ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

ST_MultiPolygon

Se *l'srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_MultiPolygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_MultiPolygon per creare ed inserire un multipoligono dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un multipoligono nel sistema di riferimento spaziale 1. Il multipoligono appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono:

- Poligono 1: (3, 3) (4, 6) (5, 3) (3, 3)
- Poligono 2: (8, 24) (9, 25) (1, 28) (8, 24)
- Poligono 3: (13, 33) (7, 36) (1, 40) (10, 43) (13, 33)

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpolys (id INTEGER, geometry ST_MultiPolygon)
```

```
INSERT INTO sample_mpolys
VALUES (1110,
       ST_MultiPolygon ('multipolygon (( (3 3, 4 6, 5 3, 3 3),
                                         (8 24, 9 25, 1 28, 8 24),
                                         (13 33, 7 36, 1 40, 10 43 13 33) ))', 1) )
```

La seguente istruzione SELECT restituisce il multipoligono registrato nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(350) ) MULTI_POLYGON
FROM sample_mpolys
WHERE id = 1110
```

Risultati:

```
ID          MULTI_POLYGON
-----
1110 MULTIPOLYGON ((( 13.00000000 33.00000000, 10.00000000 43.00000000,
1.00000000 40.00000000, 7.00000000 36.00000000,
13.00000000 33.00000000)),
(( 8.00000000 24.00000000, 9.00000000 25.00000000,
1.00000000 28.00000000, 8.00000000 24.00000000)),
(( 3.00000000 3.00000000, 4.00000000 6.00000000,
5.00000000 3.00000000, 3.00000000 3.00000000)))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_NumGeometries

ST_NumGeometries utilizza un insieme di forme geometriche come parametro di input e restituisce il numero di forme contenute nell'insieme.

Se l'insieme di forme geometriche dato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_NumGeometries—(—insieme—)—————►►
```

Parametro:

insieme

Un valore di tipo ST_GeomCollection o uno dei tipi secondari che rappresenta l'insieme di forme geometriche di cui viene restituito il numero di forme che lo compongono.

Tipo restituito:

INTEGER

Esempio:

Nella tabella SAMPLE_GEOMCOLL sono memorizzati due insiemi di forme geometriche. Uno è un multipoligono, l'altro un multipunto. La funzione ST_NumGeometries stabilisce quante singole forme geometriche sono contenute in ciascun insieme di forme.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geomcoll (id INTEGER, geometry ST_GeomCollection)

INSERT INTO sample_geomcoll
VALUES( 1,
        ST_MultiPolygon ('multipolygon (( (3 3, 4 6, 5 3, 3 3),
                                         (8 24, 9 25, 1 28, 8 24),
                                         (13 33, 7 36, 1 40, 10 43, 13 33) ))', 1) )

INSERT INTO sample_geomcoll
VALUES (2, ST_MultiPoint ('multipoint (1 2, 4 3, 5 6, 7 6, 8 8)', 1) )

SELECT id, ST_NumGeometries (geometry) NUM_GEOMS_IN_COLL
FROM sample_geomcoll
```

Risultati:

ID	NUM_GEOMS_IN_COLL
1	3
2	5

Riferimenti correlati:

- “ST_GeometryN” a pagina 395

ST_NumInteriorRing

ST_NumInteriorRing utilizza un poligono come parametro di input e restituisce il numero delle circonferenze interne al poligono.

Se il poligono dato è nullo o vuoto, viene restituito un valore nullo.

Se il poligono non contiene circonferenze interne, viene restituito 0 (zero).

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_NumInteriorRing(—poligono—) ◀◀
```

Parametro:

poligono

Un valore di tipo ST_Polygon che rappresenta il poligono di cui viene restituito il numero di circonferenze interne.

Tipo restituito:

INTEGER

Esempio:

L'esempio seguente consente di creare due poligoni:

- Uno con due circonferenze interne.
- Uno senza alcuna circonferenza.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1, ST_Polygon('polygon
((40 120, 90 120, 90 150, 40 150, 40 120),
(50 130, 60 130, 60 140, 50 140, 50 130),
(70 130, 80 130, 80 140, 70 140, 70 130))', 0) )
```

```
INSERT INTO sample_polys
VALUES (2, ST_Polygon('polygon ((5 15, 50 15, 50 105, 5 15))', 0) )
```

Per restituire il numero di circonferenze contenuto nelle forme geometriche della tabella, viene utilizzata la funzione ST_NumInteriorRing:

```
SELECT id, ST_NumInteriorRing(geometry) NUM_RINGS
FROM sample_polys
```

Risultati:

ID	NUM_RINGS
1	2
2	0

Riferimenti correlati:

- “ST_InteriorRingN” a pagina 402

ST_NumLineStrings

ST_NumLineStrings utilizza una multilinea come parametro di input e restituisce il numero delle linee che la compongono.

Se la multilinea data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_NumLineStrings—(—*multilinea*—)—————►◄

Parametro:

multilinea

Un valore di tipo ST_MultiLineString che rappresenta la multilinea di cui vengono calcolate le linee che la costituiscono.

Tipo restituito:

INTEGER

Esempio:

Le multilinee vengono memorizzate nella tabella SAMPLE_MLINES. La funzione ST_NumLineStrings consente di stabilire quante singole forme geometriche sono contenute in ciascuna multilinea.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mlines (id INTEGER, geometry ST_MultiLineString)
```

```
INSERT INTO sample_mlines
VALUES (110, ST_MultiLineString ('multilinestring
( (33 2, 34 3, 35 6),
(28 4, 29 5, 31 8, 43 12),
(39 3, 37 4, 36 7))', 1) )
```

```
INSERT INTO sample_mlines
VALUES (111, ST_MultiLineString ('multilinestring
( (3 2, 4 3, 5 6),
(8 4, 9 5, 3 8, 4 12))', 1) )
```

```
SELECT id, ST_NumLineStrings (geometry) NUM_WITHIN
FROM sample_mlines
```

Risultati:

ID	NUM_WITHIN
110	3
111	2

Riferimenti correlati:

- “ST_LineStringN” a pagina 420

ST_NumPoints

ST_NumPoints utilizza una forma geometrica come parametro di input e restituisce il numero di punti utilizzati per definire la forma. Ad esempio, se la forma geometrica è un poligono definito da 5 punti, il numero restituito sarà 5.

Se la forma geometrica data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_NumPoints—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui viene restituito il numero di punti.

Tipo restituito:

INTEGER

Esempio:

Nella tabella sono memorizzate diverse forme geometriche. La funzione ST_NumPoints determina quanti punti sono contenuti in ciascuna forma geometrica nella tabella SAMPLE_GEOMETRIES.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (spatial_type VARCHAR(18), geometry ST_Geometry)
```

```
INSERT INTO sample_geometries
VALUES ('st_point',
       ST_Point (2, 3, 0) )
```

```
INSERT INTO sample_geometries
VALUES ('st_linestring',
       ST_LineString ('linestring (2 5, 21 3, 23 10)', 0) )
```

```
INSERT INTO sample_geometries
VALUES ('st_polygon',
       ST_Polygon ('polygon ((110 120, 110 140, 120 130, 110 120))', 0) )
```

```
SELECT spatial_type, ST_NumPoints (geometry) NUM_POINTS
FROM sample_geometries
```

Risultati:

SPATIAL_TYPE	NUM_POINTS
st_point	1
st_linestring	3
st_polygon	4

Riferimenti correlati:

- “ST_PointN” a pagina 471

ST_NumPolygons

ST_NumPolygons utilizza un multipoligono come parametro di input e restituisce il numero di poligoni in esso contenuti.

Se il multipoligono dato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_NumPolygons—(—multipoligono—)—————►►

Parametro:

multipoligono

Un valore di tipo ST_MultiPolygon che rappresenta il multipoligono di cui vengono calcolati i multipoligoni che lo costituiscono.

Tipo restituito:

INTEGER

Esempio:

I multipoligoni vengono memorizzati nella tabella SAMPLE_MPOLYS. La funzione ST_NumPolygons consente di stabilire quante singole forme geometriche sono contenute in ciascun multipoligono.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpolys (id INTEGER, geometry ST_MultiPolygon)

INSERT INTO sample_mpolys
VALUES( 1,
        ST_MultiPolygon ('multipolygon (( (3 3, 4 6, 5 3, 3 3),
                                         (8 24, 9 25, 1 28, 8 24),
                                         (13 33, 7 36, 1 40, 10 43, 13 33) ))', 1) )

INSERT INTO sample_polys
VALUES(2,
        ST_MultiPolygon ('multipolygon empty', 1) )

INSERT INTO sample_polys
VALUES (3,
        ST_MultiPolygon ('multipolygon (( (3 3, 4 6, 5 3, 3 3),
                                         (13 33, 7 36, 1 40, 10 43, 13 33) ))', 1) )

SELECT id, ST_NumPolygons (geometry) NUM_WITHIN
FROM sample_mpolys
```

Risultati:

ID	NUM_WITHIN
1	3
2	0
3	2

Riferimenti correlati:

- “ST_PolygonN” a pagina 478

ST_Overlaps

ST_Overlaps utilizza due forme geometriche come parametri di input e restituisce 1 se l'intersezione delle forme è rappresentata da una forma geometrica di stesse dimensioni ma non uguale alle forme date. In caso contrario, restituisce 0 (zero).

Se una delle due forme geometriche è nulla o vuota, viene restituito un valore nullo.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Sintassi:

```
►►—db2gse.ST_Overlaps—(—forma geometrica 1—,—forma geometrica 2—)————►◄
```

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da confrontare alla *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da confrontare alla *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_Overlaps. Diverse forme geometriche vengono create e inseriti nella tabella SAMPLE_GEOMETRIES

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Point (10, 20, 1)),
      (2, ST_Point ('point (41 41)', 1) ),
      (10, ST_LineString ('linestring (1 10, 3 12, 10 10)', 1) ),
      (20, ST_LineString ('linestring (50 10, 50 12, 45 10)', 1) ),
      (30, ST_LineString ('linestring (50 12, 50 10, 60 8)', 1) ),
      (100, ST_Polygon ('polygon ((0 0, 0 40, 40 40, 40 0, 0 0))', 1) ),
      (110, ST_Polygon ('polygon ((30 10, 30 30, 50 30, 50 10, 30 10))', 1) ),
      (120, ST_Polygon ('polygon ((0 50, 0 60, 40 60, 40 50))', 1) )
```

Esempio 1:

Questo esempio consente di individuare gli ID dei punti che si sovrappongono.

```
SELECT sg1.id, sg2.id
CASE ST_Overlaps (sg1.geometry, sg2.geometry)
WHEN 0 THEN 'Points_do_not_overlap'
WHEN 1 THEN 'Points_overlap'
END
AS OVERLAP
FROM sample_geometries sg1, sample_geometries sg2
WHERE sg1.id < 10 AND sg2.id < 10 AND sg1.id >= sg2.id
```

Risultati:

ID	ID	OVERLAP
	1	1 Points_do_not_overlap
	2	1 Points_do_not_overlap
	2	2 Points_do_not_overlap

Esempio 2:

Questo esempio consente di individuare gli ID delle linee che si sovrappongono.

```
SELECT sg1.id, sg2.id
  CASE ST_Overlaps (sg1.geometry, sg2.geometry)
    WHEN 0 THEN 'Lines_do_not_overlap'
    WHEN 1 THEN 'Lines_overlap'
  END
  AS OVERLAP
FROM sample_geometries sg1, sample_geometries sg2
WHERE sg1.id >= 10 AND sg1.id < 100
  AND sg2.id >= 10 AND sg2.id < 100
  AND sg1.id >= sg2.id
```

Risultati:

ID	ID	OVERLAP
	10	10 Lines_do_not_overlap
	20	10 Lines_do_not_overlap
	30	10 Lines_do_not_overlap
	20	20 Lines_do_not_overlap
	30	20 Lines_overlap
	30	30 Lines_do_not_overlap

Esempio 3:

Questo esempio consente di individuare gli ID dei poligoni che si sovrappongono.

```
SELECT sg1.id, sg2.id
  CASE ST_Overlaps (sg1.geometry, sg2.geometry)
    WHEN 0 THEN 'Polygons_do_not_overlap'
    WHEN 1 THEN 'Polygons_overlap'
  END
  AS OVERLAP
FROM sample_geometries sg1, sample_geometries sg2
WHERE sg1.id >= 100 AND sg2.id >= 100 AND sg1.id >= sg2.id
```

Risultati:

ID	ID	OVERLAP
	100	100 Polygons_do_not_overlap
	110	100 Polygons_overlap
	120	100 Polygons_do_not_overlap
	110	110 Polygons_do_not_overlap
	120	110 Polygons_do_not_overlap
	120	120 Polygons_do_not_overlap

Riferimenti correlati:

- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

ST_Perimeter

ST_Perimeter utilizza una superficie o una multisuperficie ed, eventualmente, un'unità di misura come parametri di input e restituisce il perimetro della superficie o multisuperficie, ovvero la lunghezza del contorno, misurato in base all'unità di misura specificata o predefinita.

Se la superficie o multisuperficie data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
▶▶ db2gse.ST_Perimeter (—superficie— [,—unità di misura—]) ▶▶
```

Parametri:

superficie

Un valore di tipo ST_Surface, ST_MultiSurface o uno dei tipi secondari di cui viene restituito il perimetro.

unità di misura

Un valore di tipo VARCHAR(128) che rappresenta le unità di misura in cui calcolare il perimetro. Le unità di misura supportate sono elencate nella vista del catalogo DB2GSE.ST_UNITS_OF_MEASURE.

Se il parametro *unità di misura* viene omissso, per determinare l'unità da utilizzare per misurare il perimetro, vengono applicate le seguenti regole:

- Se la *superficie* si trova in un sistema di coordinate proiettate o geocentrico, l'unità di misura lineare associata al sistema sarà quella predefinita.
- Se la *superficie* si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale, l'unità angolare associata a questo sistema di coordinate sarà utilizzata come predefinita.
- Se la *curva* si trova in un SRS geodetico, l'unità di misura predefinita sarà metri.

Limitazioni sulle conversioni delle unità: se si verifica una delle seguenti condizioni, viene restituito un errore (SQLSTATE 38SU4):

- La forma geometrica si trova in un sistema di coordinate non specificato e viene specificato il parametro *unità*.
- La forma geometrica si trova in un sistema di coordinate proiettate e viene specificata un'unità angolare.
- La forma geometrica si trova in un sistema di coordinate geografiche, ma non in un sistema di riferimento spaziale geodetico, e viene specificata un'unità lineare.
- La forma geometrica si trova in un sistema di coordinate geografiche, in un SRS geodetico, e viene specificata un'unità angolare.

Tipo restituito:

DOUBLE

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_Perimeter. Viene creato un sistema di riferimento spaziale con ID 4000 mediante un richiamo a db2se, e un poligono all'interno del sistema.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse

db2se create_srs se_bank -srsId 4000 -srsName new_york1983
  -xOffset 0 -yOffset 0 -xScale 1 -yScale 1
  -coordsysName NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

Viene creata la tabella SAMPLE_POLYS per memorizzare una forma geometrica con perimetro 18.

```
CREATE TABLE sample_polys (id SMALLINT, geometry ST_Polygon)

INSERT INTO sample_polys
  VALUES (1, ST_Polygon ('polygon ((0 0, 0 4, 5 4, 5 0, 0 0))', 4000))
```

Esempio 1:

Questo esempio consente di elencare l'ID e il perimetro del poligono.

```
SELECT id, ST_Perimeter (geometry) AS PERIMETER
  FROM sample_polys
```

Risultati:

ID	PERIMETER
1	+1.80000000000000E+001

Esempio 2:

Questo esempio consente di elencare l'ID e il perimetro calcolato in metri del poligono.

```
SELECT id, ST_Perimeter (geometry, 'METER') AS PERIMETER_METER
  FROM sample_polys
```

Risultati:

ID	PERIMETER_METER
1	+5.48641097282195E+000

ST_PerpPoints

ST_PerpPoints utilizza una curva o una multicurva e un punto come parametri di input e restituisce la proiezione perpendicolare del punto dato sulla curva o sulla multicurva. Viene restituito il punto con la distanza inferiore tra il punto dato e il punto perpendicolare. Se due o più punti proiettati perpendicolarmente sono equidistanti dal punto dato, verranno tutti restituiti. Se non è possibile creare un punto perpendicolare, viene restituito un punto vuoto.

Se la curva o la multicurva ha coordinate Z o M, le coordinate Z o M dei punti ottenuti come risultato verranno calcolati mediante l'interpolazione della curva o multicurva.

Se la curva è o il punto sono vuoti, viene restituito un punto vuoto. Se la curva è o il punto sono nulli, viene restituito un punto nullo.

ST_PerpPoints

La funzione può essere richiamata come metodo.

Sintassi:

► db2gse.ST_PerpPoints(—*curva*—,—*punto*—) ◀

Parametri:

curva Un valore di tipo ST_Curve, ST_MultiCurve o uno dei tipi secondari che rappresenta la curva o la multicurva in cui viene restituita la proiezione perpendicolare del *punto*.

punto Un valore di tipo ST_Point che rappresenta il punto perpendicolare proiettato sulla *curva*.

Tipo restituito:

db2gse.ST_MultiPoint

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_PerpPoints per individuare i punti perpendicolari alla linea memorizzata nella seguente tabella. Per creare la linea, viene utilizzata la funzione ST_LineString nell'istruzione INSERT.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, line ST_LineString)

INSERT INTO sample_lines (id, line)
VALUES (1, ST_LineString('linestring (0 10, 0 0, 10 0, 10 10)' , 0) )
```

Esempio 1:

Questo esempio consente di individuare la proiezione perpendicolare sulla linea di un punto avente coordinate (5, 0). La funzione ST_AsText viene utilizzata per convertire il valore restituito (un multipunto) nella relativa rappresentazione WKT.

```
SELECT CAST ( ST_AsText( ST_PerpPoints( line, ST_Point(5, 0) ) )
AS VARCHAR(50) ) PERP
FROM sample_lines
```

Risultati:

```
PERP
-----
MULTIPOINT ( 5.00000000 0.00000000)
```

Esempio 2:

Questo esempio consente di individuare la proiezione perpendicolare sulla linea di un punto avente coordinate (5, 5). In questo caso, sulla linea esistono tre punti equidistanti dalla posizione data. Di conseguenza, viene restituito un multipunto composto dai tre punti.

```
SELECT CAST ( ST_AsText( ST_PerpPoints( line, ST_Point(5, 5) ) )
AS VARCHAR(160) ) PERP
FROM sample_lines
```

Risultati:

```
PERP
-----
MULTIPOINT ( 0.00000000 5.00000000, 5.00000000 0.00000000, 10.00000000 5.00000000)
```

Esempio 3:

Questo esempio consente di individuare la proiezione perpendicolare sulla linea di un punto avente coordinate (5, 10). In questo caso esistono tre punti perpendicolari diversi che è possibile rilevare. Tuttavia, la funzione ST_PerpPoints restituisce solo i punti più vicini al punto dato. Di conseguenza, viene restituito un multipunto composto dai due punti più vicini. Il terzo punto non viene incluso.

```
SELECT CAST ( ST_AsText( ST_PerpPoints( line, ST_Point(5, 10) ) )
AS VARCHAR(80) ) PERP
FROM sample_lines
```

Risultati:

```
PERP
-----
MULTIPOINT ( 0.00000000 10.00000000, 10.00000000 10.00000000 )
```

Esempio 4:

Questo esempio consente di individuare la proiezione perpendicolare sulla linea di un punto avente coordinate (5, 15).

```
SELECT CAST ( ST_AsText( ST_PerpPoints( line, ST_Point('point(5 15)', 0) ) )
AS VARCHAR(80) ) PERP
FROM sample_lines
```

Risultati:

```
PERP
-----
MULTIPOINT ( 5.00000000 0.00000000 )
```

Esempio 5:

In questo esempio, il punto specificato avente coordinate (15 15) non dispone di proiezioni perpendicolari sulla linea. Di conseguenza, viene restituita una forma geometrica vuota.

```
SELECT CAST ( ST_AsText( ST_PerpPoints( line, ST_Point(15, 15) ) )
AS VARCHAR(80) ) PERP
FROM sample_lines
```

Risultati:

```
PERP
-----
MULTIPOINT EMPTY
```

ST_Point

ST_Point genera un punto da una delle seguenti serie di input:

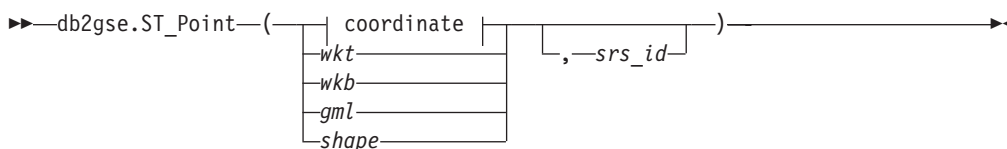
- Solo coordinate X e Y
- Coordinate X, Y e Z
- Coordinate X, Y, Z e M
- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape
- Una rappresentazione in GML (geography markup language)

Per identificare il sistema di riferimento spaziale in cui verrà inserito il punto ottenuto come risultato, è possibile specificare un identificativo di sistema.

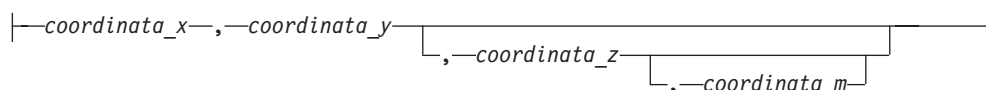
ST_Point

Se il punto viene generato dalle coordinate, e se la coordinata X o Y è nulla, si verifica una condizione di eccezione (SQLSTATE 38SUP). Se la coordinata Z o M è nulla, il punto ottenuto come risultato non avrà una delle due coordinate. Se il punto viene generato dalla propria rappresentazione WKT, WKB, shape o GML e la rappresentazione è nulla, viene restituito un valore nullo.

Sintassi:



coordinate:



Parametri:

- wkt* Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del punto ottenuto come risultato.
- wkb* Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del punto ottenuto come risultato.
- gml* Un valore di tipo CLOB(2G) che rappresenta il punto ottenuto come risultato utilizzando il linguaggio GML.
- shape* Un valore di tipo BLOB(2G) che rappresenta la rappresentazione shape del punto ottenuto come risultato.
- srs_id* Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il punto ottenuto come risultato.
Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).
Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).
- coordinata_x* Un valore di tipo DOUBLE che specifica la coordinata X del punto ottenuto come risultato.
- coordinata_y* Un valore di tipo DOUBLE che specifica la coordinata Y del punto ottenuto come risultato.
- coordinata_z* Un valore di tipo DOUBLE che specifica la coordinata Z del punto ottenuto come risultato.
Se il parametro *coordinata_z* viene omissso, il punto ottenuto come risultato non avrà coordinata Z. Per tale punto, il risultato di ST_Is3D è 0 (zero).
- coordinata_m* Un valore di tipo DOUBLE che specifica la coordinata M del punto ottenuto come risultato.

Se il parametro *coordinata_m* viene omissso, il punto ottenuto come risultato non avrà misure. Per tale punto, il risultato di ST_IsMeasured è 0 (zero).

Tipo restituito:

db2gse.ST_Point

Esempio:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Esempio 1:

Questo esempio mostra come utilizzare ST_Point per creare ed inserire punti. Il primo punto viene creato utilizzando una serie di coordinate X e Y. Il secondo punto viene creato utilizzando la rispettiva rappresentazione well-known text. Entrambi i punti sono forme geometriche del sistema di riferimento spaziale 1.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points
VALUES (1100, ST_Point (10, 20, 1) )
```

```
INSERT INTO sample_points
VALUES (1101, ST_Point ('point (30 40)', 1) )
```

La seguente istruzione SELECT restituisce i punti registrati nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(90)) POINTS
FROM sample_points
```

Risultati:

```
ID          POINTS
-----
1110 POINT ( 10.00000000 20.00000000)
1101 POINT ( 30.00000000 40.00000000)
```

Esempio 2:

L'esempio inserisce un record nella tabella SAMPLE_POINTS con ID 1103 e un valore punto con coordinata X uguale a 120, coordinata Y uguale a 358, coordinata M uguale a 34 e nessuna coordinata Z.

```
INSERT INTO SAMPLE_POINTS(ID, GEOMETRY)
VALUES(1103, db2gse.ST_Point(120, 358, CAST(NULL AS DOUBLE), 34, 1))
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(90) ) POINTS
FROM sample_points
```

Risultati:

```
ID          POINTS
-----
1103 POINT M ( 120.0000000 358.0000000 34.00000000)
```

Esempio 3:

ST_Point

L'esempio inserisce una riga nella tabella SAMPLE_POINTS con ID 1104 e un valore punto con coordinata X uguale a 1003, coordinata Y uguale a 9876, coordinata Z uguale a 20 e nel sistema di riferimento spaziale 0, utilizzando la rappresentazione GML.

```
INSERT INTO SAMPLE_POINTS(ID, GEOMETRY)
VALUES(1104, db2gse.ST_Point('<gml:Point><gml:coord>
  <gml:x>1003</gml:x><gml:y>9876</gml:y><gml:z>20</gml:z>
</gml:coord></gml:Point>', 1))
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(90) ) POINTS
FROM sample_points
```

Risultati:

```
ID          POINTS
-----
1104 POINT Z ( 1003.000000 9876.000000 20.00000000)
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_Is3d” a pagina 407
- “ST_IsMeasured” a pagina 410
- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_PointFromText

ST_PointFromText utilizza la rappresentazione WKT (well-known text) di un punto ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il punto corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_Point. La flessibilità di ST_Point consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
►► db2gse.ST_PointFromText ( ( wkt [ , srs_id ] ) ) ◀◀
```

Parametri:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del punto ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il punto ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se *l'srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Point

Esempio:

Questo esempio mostra come utilizzare ST_PointFromText per creare ed inserire un punto dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un punto nel sistema di riferimento spaziale 1. Il punto appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono: (10, 20).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points
VALUES (1110, ST_PointFromText ('point (30 40)', 1) )
```

La seguente istruzione SELECT restituisce il poligono registrato nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(35) ) POINTS
FROM sample_points
WHERE id = 1110
```

Risultati:

```
ID          POINTS
-----
1110 POINTS ( 30.00000000 40.00000000)
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_Point” a pagina 465
- “ST_PointFromWKB” a pagina 469

ST_PointFromWKB

ST_PointFromWKB utilizza la rappresentazione WKB (well-known binary) di un punto ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il punto corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_Point. La flessibilità di ST_Point consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
►► db2gse.ST_PointFromWKB ( ( wkb , srs_id ) ) ◀◀
```

ST_PointFromWKB

Parametri:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del punto ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il punto ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Point

Esempio:

Questo esempio mostra come utilizzare ST_PointFromWKB per creare un punto dalla corrispondente rappresentazione WKB. Le forme geometriche sono punti nel sistema di riferimento spaziale 1. In questo esempio, i punti vengono memorizzati nella colonna GEOMETRY della tabella SAMPLE_POLYS, quindi la colonna WKB viene aggiornata con le rispettive rappresentazioni WKB (utilizzando la funzione ST_AsBinary). Infine, per restituire i punti dalla colonna WKB, viene utilizzata la funzione ST_PointFromWKB.

La tabella SAMPLE_POINTS contiene la colonna GEOMETRY, in cui sono memorizzati i punti, e la colonna WKB, in cui è memorizzata la rappresentazione WKB dei punti.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point, wkb BLOB(32K))
```

```
INSERT INTO sample_points
VALUES (10, ST_Point ('point (44 14)', 1) ),
VALUES (11, ST_Point ('point (24 13)', 1))
```

```
UPDATE sample_points AS temporary_correlated
SET wkb = ST_AsBinary( geometry )
WHERE id = temporary_correlated.id
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_PointFromWKB per richiamare i punti dalla colonna WKB.

```
SELECT id, CAST( ST_AsText( ST_PolyFromWKB (wkb) ) AS VARCHAR(35) ) POINTS
FROM sample_points
```

Risultati:

```
ID          POINTS
-----
10 POINT ( 44.00000000 14.00000000)
11 POINT ( 24.00000000 13.00000000)
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_Point” a pagina 465

- “ST_PointFromText” a pagina 468

ST_PointN

ST_PointN utilizza una linea o un multipunto ed un indice come parametri di input e restituisce il punto della linea o del multipunto che viene identificato dall'indice. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della linea o del multipunto.

Se la linea o il multipunto dati sono nulli o vuoti, viene restituito un valore nullo. Se l'indice è minore di 1 o maggiore del numero di punti contenuti nella linea o nel multipunto, vengono restituiti un valore nullo e una condizione di avviso (SQLSTATE 01HS2).

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_PointN(—forma geometrica—, —indice—)
```

Parametri:

forma geometrica

Un valore di tipo ST_LineString o ST_MultiPoint che rappresenta la forma geometrica da cui viene restituito il punto identificato dall'*indice*.

indice Un valore di tipo INTEGER che identifica l'*ennesimo* punto restituito dalla *forma geometrica*.

Tipo restituito:

db2gse.ST_Point

Esempio:

L'esempio seguente mostra come utilizzare la funzione ST_PointN.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, line ST_LineString)

INSERT INTO sample_lines
VALUES (1, ST_LineString ('linestring (10 10, 5 5, 0 0, 10 0, 5 5, 0 10)', 0) )

SELECT id, CAST ( ST_AsText (ST_PointN (line, 2) ) AS VARCHAR(60) ) SECOND_INDEX
FROM sample_lines
```

Risultati:

ID	SECOND_INDEX
1	POINT (5.00000000 5.00000000)

Riferimenti correlati:

- “ST_Endpoint” a pagina 377
- “ST_NumPoints” a pagina 458
- “ST_StartPoint” a pagina 483

ST_PointOnSurface

ST_PointOnSurface utilizza una superficie o una multisuperficie come parametri di input e restituisce un punto interno alla superficie o multisuperficie. Il punto rappresenta il centro della superficie.

Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della superficie o della multisuperficie.

Se la superficie o multisuperficie data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_PointOnSurface—(—superficie—)—————►►
```

Parametro:

superficie

Un valore di tipo ST_Surface, ST_MultiSurface o uno dei tipi secondari che rappresenta la forma geometrica di cui viene restituito un punto sulla superficie.

Tipo restituito:

db2gse.ST_Point

Esempio:

Nell'esempio seguente, vengono creati due poligoni e successivamente viene utilizzata ST_PointOnSurface. Uno dei poligoni ha un foro nel centro. I punti restituiti si trovano sulla superficie dei poligoni, e non si trovano necessariamente nel centro esatto.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

INSERT INTO sample_polys
VALUES( 1,
        ST_Polygon ('polygon ( (40 120, 90 120, 90 150, 40 150, 40 120) ,
                               (50 130, 80 130, 80 140, 50 140, 50 130) )' ,0) )
INSERT INTO sample_polys
VALUES(2,
        ST_Polygon ('polygon ( (10 10, 50 10, 10 30, 10 10) )', 0) )

SELECT id, CAST (ST_AsText (ST_PointOnSurface (geometry) ) AS VARCHAR(80) )
        POINT_ON_SURFACE
FROM sample_polys
```

Risultati:

```
ID          POINT_ON_SURFACE
-----
1 POINT ( 65.00000000 125.00000000)
2 POINT ( 30.00000000 15.00000000)
```

ST_PolyFromText

ST_PolyFromText utilizza la rappresentazione WKT (well-known text) di un poligono ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il poligono corrispondente.

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_Polygon. La flessibilità di ST_Polygon consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
db2gse.ST_PolyFromText( ( wkt [, srs_id ] ) )
```

Parametri:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT del poligono ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il poligono ottenuto come risultato.

Se il parametro *srs_id* viene omissso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Polygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_PolyFromText per creare ed inserire un poligono dalla corrispondente rappresentazione WKT. Il record inserito ha ID = 1110, e la forma geometrica sarà un poligono nel sistema di riferimento spaziale 1. Il poligono appartiene alla rappresentazione WKT. Le coordinate X e Y della forma geometrica sono: (50, 20) (50, 40) (70, 30).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1110, ST_PolyFromText ('polygon ((50 20, 50 40, 70 30, 50 20))', 1) )
```

La seguente istruzione SELECT restituisce il poligono registrato nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(120) ) POLYGON
FROM sample_polys
WHERE id = 1110
```

ST_PolyFromText

Risultati:

```
ID          POLYGON
-----
1110 POLYGON (( 50.00000000 20.00000000, 70.00000000 30.00000000,
                50.00000000 40.00000000, 50.00000000 20.00000000))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_PolyFromWKB” a pagina 474
- “ST_Polygon” a pagina 475

ST_PolyFromWKB

ST_PolyFromWKB utilizza la rappresentazione WKB (well-known binary) di un poligono ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input e restituisce il poligono corrispondente.

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

La funzione consigliata per ottenere lo stesso risultato è ST_Polygon. La flessibilità di ST_Polygon consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

```
db2gse.ST_PolyFromWKB(wkb [, srs_id])
```

Parametri:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB del poligono ottenuto come risultato.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per il poligono ottenuto come risultato.

Se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo numerico 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Polygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare ST_PolyFromWKB per creare un poligono dalla corrispondente rappresentazione WKB. La forma geometrica è un poligono

del sistema di riferimento spaziale 1. In questo esempio, il poligono viene memorizzato con ID = 1115 nella colonna GEOMETRY della tabella SAMPLE_POLYS, quindi la colonna WKB viene aggiornata col la rispettiva rappresentazione WKB (utilizzando la funzione ST_AsBinary). Infine, per restituire il poligono dalla colonna WKB, viene utilizzata la funzione ST_PolyFromWKB. Le coordinate X e Y della forma geometrica sono: (50, 20) (50, 40) (70, 30).

La tabella SAMPLE_POLYS contiene la colonna GEOMETRY, in cui è memorizzato il poligono, e la colonna WKB, in cui è memorizzata la rappresentazione WKB del poligono.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon,
    wkb BLOB(32K))

INSERT INTO sample_polys
    VALUES (10, ST_Polygon ('polygon ((50 20, 50 40, 70 30, 50 20))', 1) )

UPDATE sample_polys AS temporary_correlated
    SET wkb = ST_AsBinary( geometry )
    WHERE id = temporary_correlated.id
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_PolyFromWKB per richiamare il poligono dalla colonna WKB.

```
SELECT id, CAST( ST_AsText( ST_PolyFromWKB (wkb) )
    AS VARCHAR(120) ) POLYGON
    FROM sample_polys
    WHERE id = 1115
```

Risultati:

```
ID          POLYGON
-----
1115 POLYGON (( 50.00000000 20.00000000, 70.00000000
    30.00000000,50.00000000 40.00000000, 50.00000000
    20.00000000))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_PolyFromText” a pagina 473
- “ST_Polygon” a pagina 475

ST_Polygon

ST_Polygon consente di creare un poligono da uno dei seguenti input:

- Una linea chiusa che definisce la circonferenza esterna del poligono ottenuto come risultato
- Una rappresentazione WKT (well-known text)
- Una rappresentazione WKB (well-known binary)
- Una rappresentazione shape
- Una rappresentazione in GML (geography markup language)

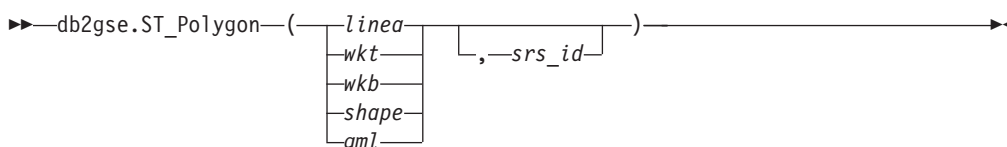
Per identificare il sistema di riferimento spaziale in cui verrà inserito il poligono ottenuto come risultato, è possibile specificare un identificativo di sistema.

ST_Polygon

Se il poligono viene generato da una linea e la linea data è nulla, viene restituito un valore nullo. Se la linea data è vuota, viene restituito un poligono vuoto. Se il poligono viene generato dalla propria rappresentazione WKT, WKB, shape o GML e la rappresentazione è nulla, viene restituito un valore nullo.

Questa funzione può essere richiamata come metodo solo nei seguenti casi: `ST_Polygon(linea)` e `ST_Polygon(linea, srs_id)`.

Sintassi:



Parametri:

- linea* Un valore di tipo `ST_LineString` che rappresenta la linea che definisce la circonferenza esterna del contorno. Se la *linea* non è chiusa e semplice, viene generata una condizione di errore (SQLSTATE 38SSL).
- wkt* Un valore di tipo `CLOB(2G)` che contiene la rappresentazione WKT del poligono ottenuto come risultato.
- wkb* Un valore di tipo `BLOB(2G)` che contiene la rappresentazione WKB del poligono ottenuto come risultato.
- shape* Un valore di tipo `BLOB(2G)` che rappresenta la rappresentazione shape del poligono ottenuto come risultato.
- gml* Un valore di tipo `CLOB(2G)` che rappresenta il poligono ottenuto come risultato utilizzando il linguaggio GML.
- srs_id* Un valore di tipo `INTEGER` che identifica il sistema di riferimento spaziale per il poligono ottenuto come risultato.

Se il poligono viene generato da un parametro *linea* dato e il parametro *srs_id* viene omesso, verrà utilizzato implicitamente il sistema di riferimento spaziale della *linea*. Diversamente, se il parametro *srs_id* viene omesso, verrà utilizzato il sistema di riferimento spaziale con identificativo 0 (zero).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo `DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS`, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

`db2gse.ST_Polygon`

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare `ST_Polygon` per creare ed inserire poligoni. Vengono creati ed inseriti tre poligoni. Tutti sono forme geometriche appartenenti al sistema di riferimento spaziale 1.

- Il primo poligono viene creato da una circonferenza (linea semplice e chiusa). Le coordinate X e Y del poligono sono: (10, 20) (10, 40) (20, 30).
- Il secondo poligono viene creato utilizzando la rispettiva rappresentazione well-known text. Le coordinate X e Y del poligono sono: (110, 120) (110, 140) (120, 130).
- Il terzo poligono è un poligono donut. Un poligono donut è formato da un poligono interno e uno esterno. Il poligono donut viene creato utilizzando la rispettiva rappresentazione WKT. Le coordinate X e Y del poligono esterno sono: (110, 120) (110, 140) (130, 140) (130, 120) (110, 120). Le coordinate X e Y del poligono interno sono: (115, 125) (115, 135) (125, 135) (125, 125) (115, 125).

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_polys
VALUES (1100,
       ST_Polygon (ST_LineString ('linestring
                                (10 20, 10 40, 20 30, 10 20)',1), 1))
```

```
INSERT INTO sample_polys
VALUES (1101,
       ST_Polygon ('polygon
                   ((110 120, 110 140, 120 130, 110 120))', 1))
```

```
INSERT INTO sample_polys
VALUES (1102,
       ST_Polygon ('polygon
                   ((110 120, 110 140, 130 140, 130 120, 110 120),
                    (115 125, 115 135, 125 135, 125 125, 115 125))', 1))
```

La seguente istruzione SELECT restituisce i poligoni registrati nella tabella:

```
SELECT id, CAST( ST_AsText( geometry ) AS VARCHAR(120) ) POLYGONS
FROM sample_polys
```

Risultati:

ID	POLYGONS
1100	POLYGON ((10.00000000 20.00000000, 20.00000000 30.00000000 10.00000000 40.00000000, 10.00000000 20.00000000))
1101	POLYGON ((110.00000000 120.00000000, 120.00000000 130.00000000 110.00000000 140.00000000, 110.00000000 120.00000000))
1102	POLYGON ((110.00000000 120.00000000, 130.00000000 120.00000000 130.00000000 140.00000000, 110.00000000 140.00000000 110.00000000 120.00000000), (115.00000000 125.00000000, 115.00000000 135.00000000 125.00000000 135.00000000, 125.00000000 135.00000000 115.00000000 125.00000000))

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “Rappresentazione WKT (well-known text)” a pagina 515
- “Rappresentazione WKB (well-known binary)” a pagina 520
- “Rappresentazione shape” a pagina 522
- “Rappresentazione GML (Geography Markup Language)” a pagina 522

ST_PolygonN

ST_PolygonN utilizza un multipoligono ed un indice come parametri di input e restituisce il poligono identificato dall'indice. Il poligono ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale del multipoligono dato.

Se il multipoligono ha valore nullo o è vuoto, o se l'indice è minore di 1 o maggiore del numero di poligoni, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_PolygonN(—multipoligono—,—indice—)◄◄
```

Parametri:

multipoligono

Un valore di tipo ST_MultiPolygon che rappresenta il multipoligono da cui viene restituito il poligono identificato dall'*indice*.

indice

Un valore di tipo INTEGER che identifica l'*ennesimo* poligono restituito dall'*multipoligono*.

Tipo restituito:

db2gse.ST_Polygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_PolygonN.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_mpolys (id INTEGER, geometry ST_MultiPolygon)

INSERT INTO sample_mpolys
VALUES (1, ST_Polygon ('multipolygon (((3 3, 4 6, 5 3, 3 3),
                               (8 24, 9 25, 1 28, 8 24)
                               (13 33, 7 36, 1 40, 10 43,
                               13 33)))', 1))

SELECT id, CAST ( ST_AsText (ST_PolygonN (geometry, 2) )
AS VARCHAR(120) ) SECOND_INDEX
FROM sample_mpolys
```

Risultati:

```
ID          SECOND_INDEX
-----
1 POLYGON (( 8.00000000 24.00000000, 9.00000000 25.00000000,
             1.00000000 28.00000000, 8.00000000 24.00000000))
```

Riferimenti correlati:

- "ST_NumPolygons" a pagina 459

ST_Relate

ST_Relate utilizza due forme geometriche e una matrice DE-9IM (Dimensionally Extended 9 Intersection Model) come parametri di input e se le geometrie soddisfano le condizioni specificate dalla matrice, restituisce 1. In caso contrario, restituisce 0 (zero).

Se una delle forme geometriche date è nulla o vuota, viene restituito un valore nullo.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_Relate—————►
►—(—forma geometrica 1—,—forma geometrica 2—,—matrice—)————►
```

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da confrontare alla *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da confrontare alla *forma geometrica 1*.

matrice Un valore di tipo CHAR(9) che rappresenta la matrice DE-9IM da utilizzare per confrontare la *forma geometrica 1* e la *forma geometrica 2*.

Tipo restituito:

INTEGER

Esempio:

Il codice seguente consente di creare due poligoni separati. Successivamente, viene utilizzata la funzione ST_Relate per determinare le diverse relazioni tra i due poligoni. Ad esempio, se i due poligoni si sovrappongono.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_polys (id INTEGER, geometry ST_Polygon)

INSERT INTO sample_polys
VALUES( 1,
       ST_Polygon('polygon ( (40 120, 90 120, 90 150, 40 150, 40 120) )', 0))
INSERT INTO sample_polys
VALUES(2,
       ST_Polygon('polygon ( (30 110, 50 110, 50 130, 30 130, 30 110) )', 0))

SELECT ST_Relate(a.geometry, b.geometry, 'T***T**') "Overlaps ",
       ST_Relate(a.geometry, b.geometry, 'T*T***FF*') "Contains ",
       ST_Relate(a.geometry, b.geometry, 'T*F**F***') "Within ",
       ST_Relate(a.geometry, b.geometry, 'T*****') "Intersects",
       ST_Relate(a.geometry, b.geometry, 'T*F***FF2') "Equals "
FROM sample_polys a, sample_polys b
WHERE a.id = 1 AND b.id = 2
```

ST_Relate

Risultati:

Overlaps	Contains	Within	Intersects	Equals
-----	-----	-----	-----	-----
1	0	0	1	0

Riferimenti correlati:

- “Funzioni che confrontano le funzioni geografiche” a pagina 304

ST_RemovePoint

ST_RemovePoint utilizza una curva ed un punto come parametri di input e restituisce la curva data dopo aver rimosso da essa tutti i punti uguali al punto dato. Se la curva ha coordinate Z o M, il punto dovrà avere le stesse coordinate. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

Se la curva è vuota, viene restituita una curva nulla. Se la curva o il punto dato hanno valore nullo o sono vuoti, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►► db2gse.ST_RemovePoint(—*curva*—, —*punto*—) ◀◀

Parametri:

- curva* Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la curva da cui viene rimosso il *punto*.
- punto* Un valore di tipo ST_Point che rappresenta il punto da rimuovere dalla *curva*.

Tipo restituito:

db2gse.ST_Curve

Esempi:

Nell'esempio seguente, vengono aggiunte due linee alla tabella SAMPLE_LINES. Tali linee vengono utilizzate negli altri esempi.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, line ST_LineString)
```

```
INSERT INTO sample_lines
VALUES (1, ST_LineString('linestring
(10 10, 5 5, 0 0, 10 0, 5 5, 0 10)', 0))
```

```
INSERT INTO sample_lines
VALUES (2, ST_LineString('linestring z
(0 0 4, 5 5 5, 10 10 6, 5 5 7, 0 0 8)', 0))
```

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Esempio 1:

Il seguente esempio consente di eliminare il punto (5, 5) dalla linea con ID = 1. Il punto si presenta due volte nella linea e viene rimosso in entrambe le ricorrenze.

```
SELECT CAST(ST_AsText (ST_RemovePoint (line, ST_Point(5, 5) ) )
AS VARCHAR(120) ) RESULT
FROM sample_lines
WHERE id = 1
```

Risultati:

RESULT

```
-----
LINESTRING ( 10.00000000 10.00000000, 0.00000000 0.00000000,
            10.00000000 0.00000000, 0.00000000 10.00000000)
```

Esempio 2:

Il seguente esempio consente di eliminare il punto (5, 5, 5) dalla linea con ID = 2. Il punto si presenta una sola volta, quindi viene eliminata solo una ricorrenza.

```
SELECT CAST (ST_AsText (ST_RemovePoint (line, ST_Point(5.0, 5.0, 5.0)))
AS VARCHAR(160) ) RESULT
FROM sample_lines
WHERE id=2
```

Risultati:

RESULT

```
-----
LINESTRING Z ( 0.00000000 0.00000000 4.00000000, 10.00000000 10.00000000
            6.00000000, 5.00000000 5.00000000 7.00000000, 0.00000000
            0.00000000 8.00000000)
```

ST_SrsId, ST_SRID

ST_SrsId (o ST_SRID) utilizza una forma geometrica ed, eventualmente, un identificativo di sistema di riferimento spaziale come parametri di input. Gli elementi restituiti dipendono dai parametro di input specificati:

- Se viene indicato l'identificativo del sistema di riferimento spaziale, viene restituita la forma geometrica il cui sistema di riferimento spaziale è stato modificato nel sistema specificato. La forma geometrica non verrà trasformata.
- Se non viene indicato alcun identificativo di sistema di riferimento spaziale, viene restituito l'identificativo della forma geometrica data.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

Le funzioni possono essere richiamate come metodi.

Sintassi:

```
db2gse.ST_SrsId (—forma geometrica [,—srs_id] )
```

Parametri:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica per la quale bisognerà impostare o restituire l'identificativo del sistema di riferimento spaziale.

ST_SrsId e ST_SRID

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale da utilizzare per la forma geometrica ottenuta come risultato.

Attenzione: Se il parametro viene specificato, la forma geometrica non viene trasformata, ma viene restituita con sistema di riferimento spaziale modificato nel sistema specificato. Di conseguenza, alcuni dati potrebbero risultare danneggiati. Per le trasformazioni, utilizzare ST_Transform.

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipi restituiti:

- INTEGER, se non viene specificato un *srs_id*
- db2gse.ST_Geometry, se l'*srs_id* viene specificato

Esempio:

Vengono creati due punti in due sistemi di riferimento spaziali diversi tra loro. L'ID del sistema di riferimento spaziale associato a ciascun punto può essere individuato mediante la funzione ST_SrsId.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points
VALUES (1, ST_Point( 'point (80 180)', 0 ) )
INSERT INTO sample_points
VALUES (2, ST_Point( 'point (-74.21450127 + 42.03415094)', 1 ) )
```

```
SELECT id, ST_SRSID (geometry) SRSID
FROM sample_points
```

Risultati:

ID	SRSID
1	0
2	1

Riferimenti correlati:

- “ST_Transform” a pagina 495

ST_SrsName

ST_SrsName utilizza una forma geometrica come parametro di input e restituisce il nome del sistema di riferimento spaziale in cui la forma viene rappresentata.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_SrsName(—forma geometrica—) ◀◀
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui viene restituito il nome del sistema di riferimento spaziale.

Tipo restituito:

VARCHAR(128)

Esempio:

Vengono creati due punti in sistemi di riferimento spaziali diversi tra loro. La funzione ST_SrsName viene utilizzata per individuare il nome del sistema di riferimento spaziale associato a ciascun punto.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry, ST_Point)
```

```
INSERT INTO sample_points
VALUES (1, ST_Point ('point (80 180)', 0) )
```

```
INSERT INTO sample_points
VALUES (2, ST_Point ('point (-74.21450127 + 42.03415094)', 1) )
```

```
SELECT id, ST_SrsName (geometry) SRSNAME
FROM sample_points
```

Risultati:

ID	SRSNAME
1	DEFAULT_SRS
2	NAD83_SRS_1

Riferimenti correlati:

- “ST_SrsId, ST_SRID” a pagina 481

ST_StartPoint

ST_StartPoint utilizza una curva come parametro di input e restituisce il primo punto della curva. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della curva data. Il risultato è equivalente a quello ottenuto richiamando la funzione ST_PointN(*curva*, 1)

Se la curva data è nulla o vuota, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_StartPoint(—curva—) ◀◀
```

Parametri:

curva Un valore di tipo ST_Curve o uno dei tipi secondari che rappresenta la forma geometrica da cui proviene il primo punto restituito.

Tipo restituito:

ST_StartPoint

db2gse.ST_Point

Esempio:

Nell'esempio seguente, vengono aggiunte due linee alla tabella SAMPLE_LINES. La prima linea ha coordinate X e Y. La seconda ha coordinate X, Y e Z. Per restituire il primo punto di ciascuna linea, viene utilizzata la funzione ST_StartPoint.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_lines (id INTEGER, line ST_LineString)

INSERT INTO sample_lines
VALUES (1, ST_LineString ('linestring
(10 10, 5 5, 0 0, 10 0, 5 5, 0 10)', 0))

INSERT INTO sample_lines
VALUES (1, ST_LineString ('linestring z
(0 0 4, 5 5 5, 10 10 6, 5 5 7, 0 0 8)', 0))

SELECT id, CAST( ST_AsText( ST_StartPoint( line ) ) AS VARCHAR(80))
START_POINT
FROM sample_lines
```

Risultati:

ID	START_POINT
1	POINT (10.00000000 10.00000000)
2	POINT Z (0.00000000 0.00000000 4.00000000)

Riferimenti correlati:

- “ST_Endpoint” a pagina 377
- “ST_PointN” a pagina 471

ST_SymDifference

ST_SymDifference utilizza due forme geometriche come parametri di input e restituisce la forma geometrica corrispondente alla differenza simmetrica delle due forme. La differenza simmetrica è la parte non intersecante delle due forme geometriche. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della prima forma geometrica. La dimensione della forma geometrica restituita è uguale alla dimensione delle forme geometriche di input. Entrambe le forme geometriche devono essere della stessa dimensione.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

Se le forme geometriche sono uguali, viene restituita una forma geometrica vuota di tipo ST_Point. Se la forma geometrica è nulla, viene restituito un valore nullo.

La forma geometrica ottenuta come risultato viene rappresentata nel tipo spaziale più adatto. Se possibile, verrà utilizzato un punto, una linea o un poligono, in caso contrario, verrà utilizzato un multipunto, una multilinea o un multipoligono.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_SymDifference—(—forma geometrica 1—,—forma geometrica 2—)——►►
```

Parametri:*forma geometrica 1*

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la prima forma geometrica di cui calcolare la differenza simmetrica con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la seconda forma geometrica di cui calcolare la differenza simmetrica con la *forma geometrica 1*.

Limitazioni per i dati geodetici:

- Entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.
- ST_SymDifference supporta solo i tipi di dati ST_Point, ST_Polygon, ST_MultiPoint e ST_MultiPolygon.

Tipo restituito:

db2gse.ST_Geometry

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_SymDifference. Le forme geometriche vengono memorizzate nella tabella SAMPLE_GEOMS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geoms
VALUES( 1,
       ST_Geometry ('polygon ( (10 10, 10 20, 20 20, 20 10, 10 10) )', 0))

INSERT INTO sample_geoms
VALUES
(2, ST_Geometry ('polygon ( (30 30, 30 50, 50 50, 50 30, 30 30) )', 0))

INSERT INTO sample_geoms
VALUES
(3,ST_Geometry ('polygon ( (40 40, 40 60, 60 60, 60 40, 40 40) )', 0))

INSERT INTO sample_geoms
VALUES
(4, ST_Geometry ('linestring (70 70, 80 80)' , 0) )

INSERT INTO sample_geoms
VALUES
(5, ST_Geometry('linestring(75 75, 90 90)' ,0));
```

Nei seguenti esempi, i risultati sono stati formattati nuovamente in un formato leggibile. I risultati possono variare in base al tipo di schermo.

Esempio 1:

Questo esempio utilizza la funzione ST_SymDifference per restituire la differenza simmetrica tra due poligoni distinti nella tabella SAMPLE_GEOMS.

ST_SymDifference

```
SELECT a.id, b.id,  
       CAST (ST_AsText (ST_SymDifference (a.geometry, b.geometry) )  
            AS VARCHAR(350) ) SYM_DIFF  
FROM sample_geoms a, sample_geoms b  
WHERE a.id = 1 AND b.id = 2
```

Risultati:

```
ID  ID  SYM_DIFF  
-----  
1   2  MULTIPOLYGON ((( 10.00000000 10.00000000, 20.00000000 10.00000000,  
                  20.00000000 20.00000000, 10.00000000 20.00000000,  
                  10.00000000 10.00000000)),  
              (( 30.00000000 30.00000000, 50.00000000 30.00000000,  
                  50.00000000 50.00000000, 30.00000000 50.00000000,  
                  30.00000000 30.00000000)))
```

Esempio 2:

Questo esempio utilizza la funzione ST_SymDifference per restituire la differenza simmetrica tra due poligoni intersecanti nella tabella SAMPLE_GEOMS.

```
SELECT a.id, b.id,  
       CAST (ST_AsText (ST_SymDifference (a.geometry, b.geometry) )  
            AS VARCHAR(500) ) SYM_DIFF  
FROM sample_geoms a, sample_geoms b  
WHERE a.id = 2 AND b.id = 3
```

Risultati:

```
ID  ID  SYM_DIFF  
-----  
2   3  MULTIPOLYGON ((( 40.00000000 50.00000000, 50.00000000 50.00000000,  
                  50.00000000 40.00000000, 60.00000000 40.00000000,  
                  60.00000000 60.00000000, 40.00000000 60.00000000,  
                  40.00000000 50.00000000)),  
              (( 30.00000000 30.00000000, 50.00000000 30.00000000,  
                  50.00000000 40.00000000, 40.00000000 40.00000000,  
                  40.00000000 50.00000000, 30.00000000 50.00000000,  
                  30.00000000 30.00000000)))
```

Esempio 3:

Questo esempio utilizza la funzione ST_SymDifference per restituire la differenza simmetrica tra due linee intersecanti nella tabella SAMPLE_GEOMS.

```
SELECT a.id, b.id,  
       CAST (ST_AsText (ST_SymDifference (a.geometry, b.geometry) )  
            AS VARCHAR(350) ) SYM_DIFF  
FROM sample_geoms a, sample_geoms b  
WHERE a.id = 4 AND b.id = 5
```

Risultati:

```
ID  ID  SYM_DIFF  
-----  
4   5  MULTILINESTRING (( 70.00000000 70.00000000, 75.00000000 75.00000000),  
                  ( 80.00000000 80.00000000, 90.00000000 90.00000000))
```

Riferimenti correlati:

- “ST_Difference” a pagina 365

ST_ToGeomColl

ST_ToGeomColl utilizza una forma geometrica come parametro di input e la converte in un insieme di forme geometriche. L'insieme di forme geometriche ottenute come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

Se la forma geometrica specificata è vuota, può essere di qualsiasi tipo. Successivamente, verrà convertita in ST_Multipoint, ST_MultiLineString o ST_MultiPolygon come appropriato.

Se la forma geometrica non è vuota, dovrà essere di tipo ST_Point, ST_LineString o ST_Polygon. Successivamente verranno convertite rispettivamente in ST_Multipoint, ST_MultiLineString o ST_MultiPolygon.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_ToGeomColl(—forma geometrica—)
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in un insieme di forme.

Tipo restituito:

db2gse.ST_GeomCollection

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_ToGeomColl.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Polygon ('polygon ((3 3, 4 6, 5 3, 3 3))', 1)),
       (2, ST_Point ('point (1 2)', 1))
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToGeomColl per restituire le forme geometriche e i tipi secondari corrispondenti contenuti nell'insieme.

```
SELECT id, CAST( ST_AsText( ST_ToGeomColl(geometry) )
AS VARCHAR(120) ) GEOM_COLL
FROM sample_geometries
```

Risultati:

ST_ToGeomColl

ID	GEOM_COLL
1	MULTIPOLYGON (((3.00000000 3.00000000, 5.00000000 3.00000000, 4.00000000 6.00000000, 3.00000000 3.00000000)))
2	MULTIPOINT (1.00000000 2.00000000)

ST_ToLineString

ST_ToLineString utilizza una forma geometrica come parametro di input e la converte in una linea. La linea ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota o sia una linea. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_ToLineString—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in una linea.

E' possibile convertire in linea una forma geometrica vuota o una linea. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_LineString

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_ToLineString.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('linestring z (0 10 1, 0 0 3, 10 0 5)', 0)),
(2, ST_Geometry ('point empty', 1) ),
(3, ST_Geometry ('multipolygon empty', 1) )
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToLineString per restituire le linee convertite in ST_LineString dal tipo statico ST_Geometry.

```
SELECT CAST( ST_AsText( ST_ToLineString(geometry) )
AS VARCHAR(130) ) LINES
FROM sample_geometries
```

Risultati:

LINES

```
-----
LINESTRING Z ( 0.00000000 10.00000000 1.00000000, 0.00000000
              0.00000000 3.00000000, 10.00000000 0.00000000
              5.00000000)
LINESTRING EMPTY
LINESTRING EMPTY
```

ST_ToMultiLine

ST_ToMultiLine utilizza una forma geometrica come parametro di input e la converte in una multilinea. La multilinea ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota, una linea o una multilinea. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

►►—db2gse.ST_ToMultiLine—(—*forma geometrica*—)—————►►

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in una multilinea.

E' possibile convertire in multilinea una forma geometrica vuota, una linea o una multilinea. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_MultiLineString

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_ToMultiLine.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('multilinestring ((0 10 1, 0 0 3, 10 0 5),
                                (23 43, 27 34, 35 12))', 0) ),
      (2, ST_Geometry ('linestring z (0 10 1, 0 0 3, 10 0 5)', 0) ),
      (3, ST_Geometry ('point empty', 1) ),
      (4, ST_Geometry ('multipolygon empty', 1) )
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToMultiLine per restituire le multilinee convertite in ST_MultiLineString dal tipo statico ST_Geometry.

ST_ToMultiLine

```
SELECT CAST( ST_AsText( ST_ToMultiLine(geometry) )
AS VARCHAR(130) ) LINES
FROM sample_geometries
```

Risultati:

```
LINES
-----
MULTILINESTRING Z ( 0.00000000 10.00000000 1.00000000,
                    0.00000000 0.00000000 3.00000000,
                    10.00000000 0.00000000 5.00000000)
MULTILINESTRING EMPTY
MULTILINESTRING EMPTY
```

ST_ToMultiPoint

ST_ToMultiPoint utilizza una forma geometrica come parametro di input e la converte in un multipunto. Il multipunto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota, un punto o un multipunto. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_ToMultiPoint—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in un multipunto.

E' possibile convertire in multipunto una forma geometrica vuota o che rappresenta un punto o un multipunto. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_MultiPoint

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_ToMultiPoint.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('multipoint (0 0, 0 4)', 1) ),
       (2, ST_Geometry ('point (30 40)', 1) ),
       (3, ST_Geometry ('multipolygon empty', 1) )
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToMultiPoint per restituire i multipunti convertiti in ST_MultiPoint dal tipo statico ST_Geometry.

```
SELECT CAST( ST_AsText( ST_ToMultiPoint(geometry))
AS VARCHAR(62) ) MULTIPOINTS
FROM sample_geometries
```

Risultati:

```
MULTIPOINTS
-----
MULTIPOINT ( 0.00000000 0.00000000, 0.00000000 4.00000000)
MULTIPOINT ( 30.00000000 40.00000000)
MULTIPOINT EMPTY
```

ST_ToMultiPolygon

ST_ToMultiPolygon utilizza una forma geometrica come parametro di input e la converte in un multipoligono. Il multipoligono ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota, un poligono o un multipoligono. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_ToMultiPolygon—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in un multipoligono.

E' possibile convertire in multipoligono una forma geometrica vuota o che rappresenta un poligono o un multipoligono. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_MultiPolygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio consente di creare diverse forme geometriche, quindi di utilizzare ST_ToMultiPolygon per restituire multipoligoni.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('polygon ((0 0, 0 4, 5 4, 5 0, 0 0))', 1)),
       (2, ST_Geometry ('point empty', 1)),
       (3, ST_Geometry ('multipoint empty', 1))
```

ST_ToMultiPolygon

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToMultiPolygon per restituire i multipoligono convertiti in ST_MultiPolygon dal tipo statico ST_Geometry.

```
SELECT CAST( ST_AsText( ST_ToMultiPolygon(geometry) )
AS VARCHAR(130) ) POLYGONS
FROM sample_geometries
```

Risultati:

POLYGONS

```
-----
MULTIPOLYGON (( 0.00000000 0.00000000, 5.00000000 0.00000000,
                 5.00000000 4.00000000, 0.00000000 4.00000000,
                 0.00000000 0.00000000))
```

MULTIPOLYGON EMPTY

MULTIPOLYGON EMPTY

ST_ToPoint

ST_ToPoint utilizza una forma geometrica come parametro di input e la converte in un punto. Il punto ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota o sia un punto. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_ToPoint—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in un punto.

E' possibile convertire in punto una forma geometrica vuota o che rappresenta un punto. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_Point

Esempio:

Questo esempio consente di creare tre forme geometriche in SAMPLE_GEOMETRIES e di convertire ognuna di esse in un punto.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('point (30 40)', 1) ),
       (2, ST_Geometry ('linestring empty', 1) ),
       (3, ST_Geometry ('multipolygon empty', 1) )
```

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToPoint per restituire i punti ST_Point dal tipo statico ST_Geometry.

```
SELECT CAST( ST_AsText( ST_ToPoint(geometry) ) AS VARCHAR(35) ) POINTS
FROM sample_geometries
```

Risultati:

```
POINTS
-----
POINT ( 30.00000000 40.00000000)
POINT EMPTY
POINT EMPTY
```

ST_ToPolygon

ST_ToPolygon utilizza una forma geometrica come parametro di input e la converte in un poligono. Il poligono ottenuto come risultato viene rappresentato nel sistema di riferimento spaziale della forma geometrica data.

E' necessario che la forma geometrica data sia vuota o sia un poligono. Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
►►—db2gse.ST_ToPolygon—(—forma geometrica—)—————►►
```

Parametro:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da convertire in un poligono.

E' possibile convertire in poligono una forma geometrica vuota o che rappresenta un poligono. Se non è possibile eseguire la conversione, viene generata una condizione di eccezione (SQLSTATE 38SUD).

Tipo restituito:

db2gse.ST_Polygon

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio consente di creare tre forme geometriche in SAMPLE_GEOMETRIES e di convertire ognuna di esse in un poligono.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (1, ST_Geometry ('polygon ((0 0, 0 4, 5 4, 5 0, 0 0))', 1) ),
       (2, ST_Geometry ('point empty', 1) ),
       (3, ST_Geometry ('multipolygon empty', 1) )
```

ST_ToPolygon

Nella seguente istruzione SELECT, viene utilizzata la funzione ST_ToPolygon per restituire i poligoni convertiti in ST_Polygon dal tipo statico ST_Geometry.

```
SELECT CAST( ST_AsText( ST_ToPolygon(geometry) ) AS VARCHAR(130) ) POLYGONS
FROM sample_geometries
```

Risultati:

POLYGONS

```
-----
POLYGON (( 0.00000000 0.00000000, 5.00000000 0.00000000,
           5.00000000 4.00000000,0.00000000 4.00000000,
           0.00000000 0.00000000))
```

POLYGON EMPTY

POLYGON EMPTY

ST_Touches

ST_Touches utilizza due forme geometriche come parametri di input e se le due forme si toccano, restituisce 1. In caso contrario, restituisce 0 (zero).

Le due forme geometriche si toccano se gli spazi interni non si intersecano, ma il contorno di una forma si interseca con il contorno o con lo spazio interno dell'altra.

Se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato.

Se entrambe le forme sono punti o multipunti, oppure se una delle forme è nulla o vuota, viene restituito un valore nullo.

Sintassi:

```
►►—db2gse.ST_Touches—(—forma geometrica 1—,—forma geometrica 2—)————►►
```

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina il contatto con la *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica di cui si determina il contatto con la *forma geometrica 1*.

Tipo restituito:

INTEGER

Esempio:

Diverse forme geometriche vengono aggiunte alla tabella SAMPLE_GEOMS. Successivamente, per determinare le forme geometriche in contatto l'una con l'altra, viene utilizzata la funzione ST_Touches.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry ST_Geometry)
```

```
INSERT INTO sample_geoms
```



```

VALUES (1, ST_Geometry('polygon ( (20 30, 30 30, 30 40, 20 40, 20 30) )' , 0) )
INSERT INTO sample_geoms
VALUES (2, ST_Geometry('polygon ( (30 30, 30 50, 50 50, 50 30, 30 30) )' ,0) )
INSERT INTO sample_geoms
VALUES (3, ST_Geometry('polygon ( (40 40, 40 60, 60 60, 60 40, 40 40) )' , 0) )
INSERT INTO sample_geoms
VALUES (4, ST_Geometry('linestring( 60 60, 70 70 )' , 0) )
INSERT INTO sample_geoms
VALUES (5, ST_Geometry('linestring( 30 30, 60 60 )' , 0) )
SELECT a.id, b.id, ST_Touches (a.geometry, b.geometry) TOUCHES
FROM sample_geoms a, sample_geoms b
WHERE b.id >= a.id

```

Risultati:

ID	ID	TOUCHES
1	1	0
1	2	1
1	3	0
1	4	0
1	5	1
2	2	0
2	3	0
2	4	0
2	5	1
3	3	0
3	4	1
3	5	1
4	4	0
4	5	1
5	5	0

Riferimenti correlati:

- “Funzioni che utilizzano indici per l’ottimizzazione delle interrogazioni” a pagina 122

ST_Transform

ST_Transform utilizza una forma geometrica e un identificativo di sistema di riferimento spaziale come parametri di input e trasforma la forma geometrica in modo che venga rappresentata nel sistema di riferimento dato. Vengono eseguite proiezioni e conversioni tra diversi sistemi di coordinate e le coordinate delle forme geometriche vengono regolate di conseguenza.

La forma geometrica viene convertita nel sistema di riferimento spaziale specificato solo se il sistema corrente si basa sullo stesso sistema di coordinate geografiche del sistema di riferimento specificato. Se il sistema di riferimento corrente della forma geometrica o quello specificato si basano su un sistema di coordinate proiettate, viene eseguita una proiezione inversa per stabilire il sistema di coordinate su cui si basa il sistema proiettato.

Se la forma geometrica data è nulla, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

ST_Transform

►► db2gse.ST_Transform(—forma geometrica—,—srs_id—)◄◄

Parametri:

forma geometrica

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica che verrà trasformata nel sistema di riferimento spaziale identificata dall' *srs_id*.

srs_id Un valore di tipo INTEGER che identifica il sistema di riferimento spaziale per la forma geometrica ottenuta come risultato.

Se non è possibile eseguire la trasformazione nel sistema di riferimento spaziale specificato perché il sistema spaziale corrente della *forma geometrica* non è compatibile con il sistema di riferimento spaziale identificato da *srs_id*, viene generata una condizione di errore (SQLSTATE 38SUC).

Se l'*srs_id* non identifica un sistema di riferimento spaziale elencato nella vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS, viene generata una condizione di eccezione (SQLSTATE 38SU1).

Tipo restituito:

db2gse.ST_Geometry

Esempi:

Gli esempi seguenti mostrano come utilizzare ST_Transform per convertire una forma geometrica da un sistema di riferimento spaziale in un altro.

Viene creato un sistema di riferimento spaziale piano con ID 3, mediante il seguente richiamo a db2se.

```
db2se create_srs SAMP_DB
-srsId 3 -srsName z3101a -xOffset 0 -yOffset 0 -xScale 1 -yScale 1
- coordsysName NAD_1983_StatePlane_New_York_East_FIPS_3101_Feet
```

Successivamente, vengono aggiunti i punti a:

- La tabella SAMPLE_POINTS_SP nelle coordinate piane utilizzando il sistema di riferimento spaziale.
- La tabella SAMPLE_POINTS_LL utilizzando le coordinate specificate in latitudine e longitudine.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points_sp (id INTEGER, geometry ST_Point)
CREATE TABLE sample_points_ll (id INTEGER, geometry ST_Point)

INSERT INTO sample_points_sp
VALUES (12457, ST_Point('point ( 567176.0 1166411.0)', 3) )

INSERT INTO sample_points_sp
VALUES (12477, ST_Point('point ( 637948.0 1177640.0)', 3) )

INSERT INTO sample_points_ll
VALUES (12457, ST_Point('point ( -74.22371600 42.03498700)', 1) )

INSERT INTO sample_points_ll
VALUES (12477, ST_Point('point ( -73.96293200 42.06487900)', 1) )
```

A questo punto, per convertire le forme geometriche, viene utilizzata la funzione ST_Transform.

Esempio 1:

Questo esempio consente di convertire i punti che si trovano su coordinate di latitudine e longitudine in coordinate piane.

```
SELECT id, CAST( ST_AsText( ST_Transform( geometry, 3) )
AS VARCHAR(100) ) STATE_PLANE
FROM sample_points_11
```

Risultati:

ID	STATE_PLANE
12457	POINT (567176.00000000 1166411.00000000)
12477	POINT (637948.00000000 1177640.00000000)

Esempio 2:

Questo esempio consente di convertire i punti che si trovano su coordinate piane in coordinate di latitudine e longitudine.

```
SELECT id, CAST( ST_AsText( ST_Transform( geometry, 1) )
AS VARCHAR(100) ) LAT_LONG
FROM sample_points_sp
```

Risultati:

ID	LAT_LONG
12457	POINT (-74.22371500 42.03498800)
12477	POINT (-73.96293100 42.06488000)

Riferimenti correlati:

- “Vista del catalogo DB2GSE.ST_SPATIAL_REFERENCE_SYSTEMS” a pagina 291

ST_Union

ST_Union utilizza due forme geometriche come parametri di input e restituisce la forma geometrica corrispondente all’unione delle forme date. La forma geometrica ottenuta come risultato viene rappresentata nel sistema di riferimento spaziale della prima forma geometrica.

Entrambe le forme geometriche devono essere della stessa dimensione. Se una delle due forme geometriche date è nulla, viene restituito un valore nullo.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

La forma geometrica ottenuta come risultato viene rappresentata nel tipo spaziale più adatto. Se possibile, verrà utilizzato un punto, una linea o un poligono, in caso contrario, verrà utilizzato un multipunto, una multilinea o un multipoligono.

La funzione può essere richiamata come metodo.

Sintassi:

ST_Union

►►db2gse.ST_Union(—forma geometrica 1—,—forma geometrica 2—)◄◄

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che verrà combinato alla *forma geometrica 2*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che verrà combinato alla *forma geometrica 1*.

Limitazioni per i dati geodetici: entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Tipo restituito:

db2gse.ST_Geometry

Esempi:

Le seguenti istruzioni SQL creano e riempiono la tabella SAMPLE_GEOMS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geoms (id INTEGER, geometry, ST_Geometry)
```

```
INSERT INTO sample_geoms
VALUES (1, ST_Geometry( 'polygon
((10 10, 10 20, 20 20, 20 10, 10 10) )', 0))
```

```
INSERT INTO sample_geoms
VALUES (2, ST_Geometry( 'polygon
((30 30, 30 50, 50 50, 50 30, 30 30) )', 0))
```

```
INSERT INTO sample_geoms
VALUES (3, ST_Geometry( 'polygon
((40 40, 40 60, 60 60, 60 40, 40 40) )', 0))
```

```
INSERT INTO sample_geoms
VALUES (4, ST_Geometry('linestring (70 70, 80 80)', 0))
```

```
INSERT INTO sample_geoms
VALUES (5, ST_Geometry('linestring (80 80, 100 70)', 0))
```

Nei seguenti esempi, i risultati sono stati formattati nuovamente in un formato leggibile. I risultati possono variare in base al tipo di schermo.

Esempio 1:

Questo esempio consente di individuare l'unione di due poligoni disgiunti.

```
SELECT a.id, b.id, CAST ( ST_AsText( ST_Union( a.geometry, b.geometry) )
AS VARCHAR (350) ) UNION
FROM sample_geoms a, sample_geoms b
WHERE a.id = 1 AND b.id = 2
```

Risultati:

ID	ID	UNION
1	2	MULTIPOLYGON (((10.00000000 10.00000000, 20.00000000 10.00000000, 20.00000000 20.00000000, 10.00000000 20.00000000, 10.00000000 10.00000000))

```
(( 30.00000000 30.00000000, 50.00000000
30.00000000,50.00000000 50.00000000, 30.00000000
50.00000000,30.00000000 30.00000000))
```

Esempio 2:

Questo esempio consente di individuare l'unione di due poligoni intersecanti.

```
SELECT a.id, b.id, CAST ( ST_AsText( ST_Union(a.geometry, b.geometry) )
AS VARCHAR (250)) UNION
FROM sample_geoms a, sample_geoms b
WHERE a.id = 2 AND b.id = 3
```

Risultati:

ID	ID	UNION
2	3	POLYGON ((30.00000000 30.00000000, 50.00000000 30.00000000,50.00000000 40.00000000, 60.00000000 40.00000000,60.00000000 60.00000000, 40.00000000 60.00000000 40.00000000 50.00000000, 30.00000000 50.00000000, 30.00000000 30.00000000))

Esempio 3:

Consente di individuare l'unione di due linee.

```
SELECT a.id, b.id, CAST ( ST_AsText( ST_Union( a.geometry, b.geometry) )
AS VARCHAR (250) ) UNION
FROM sample_geoms a, sample_geoms b
WHERE a.id = 4 AND b.id = 5
```

Risultati:

ID	ID	UNION
4	5	MULTILINESTRING ((70.00000000 70.00000000, 80.00000000 80.00000000), (80.00000000 80.00000000, 100.00000000 70.00000000))

ST_Within

ST_Within utilizza due forme geometriche come parametri di input e se la prima forma è contenuta completamente nella seconda, restituisce 1. In caso contrario, restituisce 0 (zero).

Se una delle forme geometriche date è nulla o vuota, viene restituito un valore nullo.

Per i dati non geodetici, se la seconda forma geometrica non è rappresentata nel sistema di riferimento spaziale della prima, viene convertita nel sistema spaziale appropriato. Per i dati geodetici, le due forme geometriche devono trovarsi nello stesso sistema di riferimento spaziale.

ST_Within esegue la stessa operazione logica di ST_Contains ma con utilizzando i parametri al contrario.

Sintassi:

```
►►—db2gse.ST_Within—(—forma geometrica 1—,—forma geometrica 2—)—►►
```

Parametri:

forma geometrica 1

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la

ST_Within

forma geometrica da analizzare per verificare se è contenuta completamente nella *forma geometrica 1*.

forma geometrica 2

Un valore di tipo ST_Geometry o uno dei tipi secondari che rappresenta la forma geometrica da analizzare per verificare se è contenuta completamente nella *forma geometrica 1*.

Limitazioni per i dati geodetici: entrambe le geometrie devono essere geodetiche e devono trovarsi nello stesso sistema di riferimento spaziale geodetico.

Tipo restituito:

INTEGER

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_Within. Vengono create le forme geometriche, quindi vengono inserite in tre tabelle, SAMPLE_POINTS, SAMPLE_LINES e SAMPLE_POLYGONS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
CREATE TABLE sample_lines (id INTEGER, line ST_LineString)
CREATE TABLE sample_polygons (id INTEGER, geometry ST_Polygon)
```

```
INSERT INTO sample_points (id, geometry)
VALUES (1, ST_Point (10, 20, 1) ),
      (2, ST_Point ('point (41 41)', 1) )
```

```
INSERT INTO sample_lines (id, line)
VALUES (10, ST_LineString ('linestring (1 10, 3 12, 10 10)', 1) ),
      (20, ST_LineString ('linestring (50 10, 50 12, 45 10)', 1) )
```

```
INSERT INTO sample_polygons (id, geometry)
VALUES (100, ST_Polygon ('polygon (( 0 0, 0 40, 40 40, 40 0, 0 0))', 1) )
```

Esempio 1:

Questo esempio consente di individuare i punti nella tabella SAMPLE_POINTS che appartengono ai poligoni della tabella SAMPLE_POLYGONS.

```
SELECT a.id POINT_ID_WITHIN_POLYGONS
FROM sample_points a, sample_polygons b
WHERE ST_Within( b.geometry, a.geometry) = 0
```

Risultati:

```
POINT_ID_WITHIN_POLYGONS
-----
2
```

Esempio 2:

Questo esempio consente di individuare le linee nella tabella SAMPLE_LINES che appartengono ai poligoni della tabella SAMPLE_POLYGONS.

```
SELECT a.id LINE_ID_WITHIN_POLYGONS
FROM sample_lines a, sample_polygons b
WHERE ST_Within( b.geometry, a.geometry) = 0
```

Risultati:

LINE_ID_WITHIN_POLYGONS

1

Riferimenti correlati:

- “ST_Contains” a pagina 359

ST_WKBToSQL

ST_WKBToSQL utilizza la rappresentazione WKB (well-known binary) di una forma geometrica e restituisce la forma corrispondente. Per la forma geometrica ottenuta come risultato, viene utilizzato il sistema di riferimento spaziale con identificativo 0 (zero).

Se la rappresentazione WKB è nulla, viene restituito un valore nullo.

ST_WKBToSQL(*wkb*) ottiene lo stesso risultato di ST_Geometry(*wkb*,0). E' preferibile utilizzare la funzione ST_Geometry invece di ST_WKBtoSQL, poiché la flessibilità di ST_Geometry consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKB.

Sintassi:

►►—db2gse.ST_WKBToSQL—(—*wkb*—)—————►►

Parametro:

wkb Un valore di tipo BLOB(2G) che contiene la rappresentazione WKB della forma geometrica ottenuta come risultato.

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio seguente mostra come utilizzare la funzione ST_WKBToSQL. Come prima attività, le forme geometriche vengono memorizzate nella colonna GEOMETRY della tabella SAMPLE_GEOMETRIES. Quindi, le rispettive rappresentazioni WKB vengono memorizzate nella colonna WKB utilizzando la funzione ST_AsBinary nell'istruzione UPDATE. Infine, per restituire le coordinate delle forme geometriche nella colonna WKB, viene utilizzata la funzione ST_WKBToSQL.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries
  (id INTEGER, geometry ST_Geometry, wkb BLOB(32K) )

INSERT INTO sample_geometries (id, geometry)
  VALUES (10, ST_Point ( 'point (44 14)', 0 ) ),
         (11, ST_Point ( 'point (24 13)', 0 ) ),
```

ST_WKBToSQL

```
(12, ST_Polygon ('polygon ((50 20, 50 40, 70 30, 50 20))', 0 ) )
UPDATE sample_geometries AS temp_correlated
  SET wkb = ST_AsBinary(geometry)
  WHERE id = temp_correlated.id
```

Per visualizzare le forme geometriche nella colonna WKB, utilizzare la seguente istruzione SELECT.

```
SELECT id, CAST( ST_AsText( ST_WKBToSQL(wkb) ) AS VARCHAR(120) ) GEOMETRIES
  FROM sample_geometries
```

Risultati:

```
ID          GEOMETRIES
-----
10 POINT ( 44.00000000 14.00000000)
11 POINT ( 24.00000000 13.00000000)
12 POLYGON (( 50.00000000 20.00000000, 70.00000000 30.00000000,
              50.00000000 40.00000000, 50.00000000 20.00000000))
```

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_Geometry” a pagina 393
- “ST_WKTToSQL” a pagina 502

ST_WKTToSQL

ST_WKTToSQL utilizza la rappresentazione WKT (well-known text) di una forma geometrica e restituisce la forma corrispondente. Per la forma geometrica ottenuta come risultato, viene utilizzato il sistema di riferimento spaziale con identificativo 0 (zero).

Se la rappresentazione WKT è nulla, viene restituito un valore nullo.

ST_WKTToSQL(*wkt*) ottiene lo stesso risultato di ST_Geometry(*wkt*,0). E' preferibile utilizzare la funzione ST_Geometry invece di ST_WKTToSQL, poiché la flessibilità di ST_Geometry consente alla funzione di utilizzare altre forme input insieme alla rappresentazione WKT.

Sintassi:

```
►► db2gse.ST_WKTToSQL(—wkt—) ◀◀
```

Parametro:

wkt Un valore di tipo CLOB(2G) che contiene la rappresentazione WKT della forma geometrica ottenuta come risultato.

Tipo restituito:

db2gse.ST_Geometry

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

L'esempio mostra come ST_WKTTToSQL è in grado di creare ed inserire forme geometriche utilizzando le corrispondenti rappresentazioni WKT.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_geometries (id INTEGER, geometry ST_Geometry)

INSERT INTO sample_geometries
VALUES (10, ST_WKTTToSQL( 'point (44 14)' ) ),
      (11, ST_WKTTToSQL ( 'point (24 13)' ) ),
      (12, ST_WKTTToSQL ('polygon ((50 20, 50 40, 70 30, 50 20))' ) )
```

La seguente istruzione SELECT restituisce le forme geometriche inserite.

```
SELECT id, CAST( ST_AsText(geometry) AS VARCHAR(120) ) GEOMETRIES
FROM sample_geometries
```

Risultati:

ID	GEOMETRIES
10	POINT (44.00000000 14.00000000)
11	POINT (24.00000000 13.00000000)
12	POLYGON ((50.00000000 20.00000000, 70.00000000 30.00000000, 50.00000000 40.00000000, 50.00000000 20.00000000))

Argomenti correlati:

- “Dati spaziali e geodetici” a pagina 4

Riferimenti correlati:

- “ST_Geometry” a pagina 393
- “ST_WKBToSQL” a pagina 501

ST_X

ST_X utilizza:

- Un punto come parametro di input e ne restituisce la coordinata X
- Un punto e la rispettiva coordinata X e restituisce lo stesso punto con la coordinata X impostata sul valore dato

Se il punto specificato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_X(—punto—, —coordinata_x—)
```

Parametri:

punto Un valore di tipo ST_Point del quale viene restituita o modificata la coordinata X.

coordinata_x

Un valore di tipo DOUBLE che rappresenta la nuova coordinata X del *punto*.

ST_X

Tipi restituiti:

- DOUBLE, se la *coordinata_x* non viene specificata
- db2gse.ST_Point, se la *coordinata_x* viene specificata

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_X. Vengono create le forme geometriche, quindi inserite nella tabella SAMPLE_POINTS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points (id, geometry)
VALUES (1, ST_Point (2, 3, 32, 5, 1) ),
       (2, ST_Point (4, 5, 20, 4, 1) ),
       (3, ST_Point (3, 8, 23, 7, 1) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata X dei punti contenuti nella tabella.

```
SELECT id, ST_X (geometry) X_COORD
FROM sample_points
```

Risultati:

ID	X_COORD
1	+2.0000000000000000E+000
2	+4.0000000000000000E+000
3	+3.0000000000000000E+000

Esempio 2:

L'esempio restituisce un punto avente la coordinata X impostata su 40.

```
SELECT id, CAST( ST_AsText( ST_X (geometry, 40)) AS VARCHAR(60) )
X_40
FROM sample_points
WHERE id=3
```

Risultati:

ID	X_40
3	POINT ZM (40.00000000 8.00000000 23.00000000 7.00000000)

Riferimenti correlati:

- "ST_M" a pagina 422
- "ST_Y" a pagina 504
- "ST_Z" a pagina 506

ST_Y

ST_Y utilizza:

- Un punto come parametro di input e ne restituisce la coordinata Y
- Un punto e la rispettiva coordinata Y e restituisce lo stesso punto con la coordinata Y impostata sul valore dato

Se il punto specificato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Y(punto [, coordinata_y])
```

Parametri:

punto Un valore di tipo ST_Point del quale viene restituita o modificata la coordinata Y.

coordinata_y
Un valore di tipo DOUBLE che rappresenta la nuova coordinata Y del *punto*.

Tipi restituiti:

- DOUBLE, se la *coordinata_y* non viene specificata
- db2gse.ST_Point, se la *coordinata_y* viene specificata

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_Y. Vengono create le forme geometriche, quindi inserite nella tabella SAMPLE_POINTS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points (id, geometry)
VALUES (1, ST_Point (2, 3, 32, 5, 1) ),
       (2, ST_Point (4, 5, 20, 4, 1) ),
       (3, ST_Point (3, 8, 23, 7, 1) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Y dei punti contenuti nella tabella.

```
SELECT id, ST_Y (geometry) Y_COORD
FROM sample_points
```

Risultati:

```
ID          Y_COORD
-----
1          +3.00000000000000E+000
2          +5.00000000000000E+000
3          +8.00000000000000E+000
```

Esempio 2:

L'esempio restituisce un punto avente la coordinata Y impostata su 40.

```
SELECT id, CAST( ST_AsText( ST_Y (geometry, 40)) AS VARCHAR(60) )
Y_40
FROM sample_points
WHERE id=3
```

Risultati:

```
ID          Y_40
-----
3 POINT ZM ( 3.00000000 40.00000000 23.00000000 7.00000000)
```

Riferimenti correlati:

- “ST_M” a pagina 422
- “ST_X” a pagina 503
- “ST_Z” a pagina 506

ST_Z

ST_Z utilizza:

- Un punto come parametro di input e ne restituisce la coordinata Z
- Un punto e una coordinata Z e restituisce il punto con la coordinata Z impostata sul valore specificato, anche se il punto non dispone di coordinata Z.

Se la coordinata Z è nulla, viene rimossa dal punto.

Se il punto specificato è nullo o vuoto, viene restituito un valore nullo.

La funzione può essere richiamata come metodo.

Sintassi:

```
db2gse.ST_Z( ( punto [, coordinata_z] ) )
```

Parametri:

punto Un valore di tipo ST_Point del quale viene restituita o modificata la coordinata Z.

coordinata_z

Un valore di tipo DOUBLE che rappresenta la nuova coordinata Z del *punto*.

Se la *coordinata_z* è nulla, la coordinata Z verrà rimossa dal *punto*.

Tipi restituiti:

- DOUBLE, se la *coordinata_z* non viene specificata
- db2gse.ST_Point, se la *coordinata_z* viene specificata

Esempi:

Gli esempi seguenti mostrano come utilizzare la funzione ST_Z. Vengono create le forme geometriche, quindi inserite nella tabella SAMPLE_POINTS.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points (id, geometry)
VALUES (1, ST_Point (2, 3, 32, 5, 1) ),
       (2, ST_Point (4, 5, 20, 4, 1) ),
       (3, ST_Point (3, 8, 23, 7, 1) )
```

Esempio 1:

Questo esempio consente di individuare la coordinata Z dei punti contenuti nella tabella.

```
SELECT id, ST_Z (geometry) Z_COORD
FROM sample_points
```

Risultati:

```
ID          Z_COORD
-----
1          +3.20000000000000E+001
2          +2.00000000000000E+001
3          +2.30000000000000E+001
```

Esempio 2:

L'esempio restituisce un punto avente la coordinata Z impostata su 40.

```
SELECT id, CAST( ST_AsText( ST_Z (geometry, 40)) AS VARCHAR(60) )
Z_40
FROM sample_points
WHERE id=3
```

Risultati:

```
ID          Z_40
-----
3 POINT ZM ( 3.00000000 8.00000000 40.00000000 7.00000000)
```

Riferimenti correlati:

- “ST_M” a pagina 422
- “ST_X” a pagina 503
- “ST_Y” a pagina 504

Aggregazione unione

Un'aggregazione unione è la combinazione delle funzioni ST_BuildUnionAggr e ST_GetAggrResult. Questa combinazione aggrega una colonna di forma geometriche di una tabella ad una singola forma geometrica, generando l'unione.

Se tutte le forme geometriche da combinare nell'unione hanno valore nullo, viene restituito un valore nullo. Se una delle forme geometriche da combinare nell'unione è nulla o vuota, verrà restituita una forma geometrica vuota di tipo ST_Point.

La funzione ST_BuildUnionAggr può anche essere richiamata come metodo.

Sintassi:

```
►► db2gse.ST_GetAggrResult ( colonna ) ►►
► MAX ( colonna ) ►►
► db2gse.ST_BuildUnionAggr ( colonna ) ►►
```

Parametri:

forme geometriche

Una colonna in una tabella contenente un tipo di ST_Geometry o un tipo secondario, che rappresenta tutte le forme geometriche da combinare in una unione.

Aggregazione unione

Tipo restituito:

db2gse.ST_Geometry

Limitazioni:

Non è possibile creare l'aggregato di una colonna spaziale in una tabella nelle seguenti situazioni:

- In ambienti MPP (massively parallel processing)
- Se nella selezione viene utilizzata una clausola GROUP BY
- Se viene utilizzata una funzione diversa dalla funzione aggregata MAX di DB2

Esempio:

Nel seguente esempio, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Questo esempio mostra come utilizzare un'aggregazione unione per unire un insieme di punti in multipunti. Diversi punti vengono aggiunti alla tabella SAMPLE_POINTS. Per creare l'unione di punti, vengono utilizzare le funzioni ST_GetAggrResult e ST_BuildUnionAggr.

```
SET CURRENT FUNCTION PATH = CURRENT FUNCTION PATH, db2gse
CREATE TABLE sample_points (id INTEGER, geometry ST_Point)
```

```
INSERT INTO sample_points
VALUES (1, ST_Point (2, 3, 1) )
INSERT INTO sample_points
VALUES (2, ST_Point (4, 5, 1) )
INSERT INTO sample_points
VALUES (3, ST_Point (13, 15, 1) )
INSERT INTO sample_points
VALUES (4, ST_Point (12, 5, 1) )
INSERT INTO sample_points
VALUES (5, ST_Point (23, 2, 1) )
INSERT INTO sample_points
VALUES (6, ST_Point (11, 4, 1) )
```

```
SELECT CAST (ST_AsText(
    ST_GetAggrResult( MAX( ST_BuildUnionAggregate (geometry) ) ))
AS VARCHAR(160)) POINT_AGGREGATE
FROM sample_points
```

Risultati:

POINT_AGGREGATE

```
-----
MULTIPOINT ( 2.00000000 3.00000000, 4.00000000 5.00000000,
             11.00000000 4.00000000, 12.00000000 5.00000000,
             13.00000000 15.00000000, 23.00000000 2.00000000)
```

Capitolo 24. Gruppi di trasformazione

Gruppi di trasformazione

Spatial Extender fornisce quattro gruppi di trasformazione che vengono utilizzati per trasferire forme geometriche tra l'applicazione server e client DB2. Questi gruppi di trasformazione possono trasferire i seguenti formati di dati:

- Rappresentazione WKT (Well-known text)
- Rappresentazione WKB (Well-known binary)
- Rappresentazione shape ESRI
- GML (Geography Markup Language)

Quando i dati vengono ricevuti da una tabella che contiene una colonna spaziale, i dati della colonna vengono trasformati in un tipo di dati CLOB(2G) o BLOB(2G), in base alle impostazioni indicate per la rappresentazione in formato binario o testo dei dati. E' anche possibile utilizzare i gruppi di trasformazione per trasferire dati spaziali al database.

Per selezionare il gruppo di trasformazione da utilizzare durante il trasferimento dei dati, utilizzare l'istruzione SET CURRENT DEFAULT TRANSFORM GROUP in modo da modificare il registro speciale DB2, n CURRENT DEFAULT TRANSFORM GROUP. DB2 utilizzerà il valore di questo registro speciale per determinare le funzioni di trasformazione da richiamare per eseguire le conversioni necessarie.

I gruppi di trasformazione possono essere utilizzati per semplificare la programmazione delle applicazioni. Invece di utilizzare esplicitamente le funzioni di conversione nelle istruzioni SQL, è possibile specificare un gruppo di trasformazione che consentirà a DB2 di svolgere l'attività.

Argomenti correlati:

- "Gruppo di trasformazione ST_WellKnownText" a pagina 509
- "Gruppo di trasformazione ST_WellKnownBinary" a pagina 510
- "Gruppo di trasformazione ST_Shape" a pagina 512
- "Gruppo di trasformazione ST_GML" a pagina 513

Gruppo di trasformazione ST_WellKnownText

Il gruppo di trasformazione ST_WellKnownText può essere utilizzato per trasmettere i dati da e a DB2[®] utilizzando la rappresentazione WKT.

Durante il trasferimento di un valore dal server dei database al client, per convertire una forma geometrica nella rappresentazione WKT, viene utilizzata la stessa funzione fornita da ST_AsText(). Durante il trasferimento della rappresentazione WKT di una forma geometrica al server dei database, viene utilizzata implicitamente la funzione ST_Geometry(CLOB) per eseguire le conversioni in un valore ST_Geometry. Utilizzando il gruppo di trasformazione per collegare i valori a DB2, le forme geometriche verranno rappresentate nel sistema di riferimento spaziale con identificativo 0 (zero).

Esempio:

ST_WellKnownText

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Esempio 1:

Il seguente script SQL mostra come utilizzare il gruppo di trasformazione ST_WellKnownText per richiamare una forma geometrica nella rispettiva rappresentazione WKT senza servirsi della funzione ST_AsText esplicita.

```
CREATE TABLE transforms_sample (  
    id INTEGER,  
    geom db2gse.ST_Geometry)  
  
INSERT  
    INTO transforms_sample  
    VALUES (1, db2gse.ST_LineString('linestring  
        (100 100, 200 100)', 0))  
  
SET CURRENT DEFAULT TRANSFORM GROUP = ST_WellKnownText  
  
SELECT id, geom  
    FROM transforms_sample  
    WHERE id = 1
```

Risultati:

```
ID  GEOM  
---  
1  LINESTRING ( 100.00000000 100.00000000, 200.00000000 100.00000000)
```

Esempio 2:

Il seguente codice C mostra come utilizzare il gruppo di trasformazione ST_WellKnownText per inserire le forme geometriche utilizzando la funzione ST_Geometry esplicita per la variabile host wkt_buffer, di tipo CLOB, che contiene la rappresentazione WKT del punto (10 10) da inserire.

```
EXEC SQL BEGIN DECLARE SECTION;  
    sqlint32 id = 0;  
    SQL TYPE IS db2gse.ST_Geometry AS CLOB(1000) wkt_buffer;  
EXEC SQL END DECLARE SECTION;  
  
// impostare il gruppo di trasformazione per tutte le istruzioni SQL  
EXEC SQL  
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_WellKnownText;  
  
id = 100;  
strcpy(wkt_buffer.data, "point ( 10 10 )");  
wkt_buffer.length = strlen(wkt_buffer.data);  
  
// inserire un punto utilizzando WKT nella colonna di tipo ST_Geometry  
EXEC SQL  
    INSERT  
    INTO transforms_sample(id, geom)  
    VALUES (:id, :wkt_buffer);
```

Gruppo di trasformazione ST_WellKnownBinary

Il gruppo di trasformazione ST_WellKnownBinary può essere utilizzato per trasmettere i dati da e a DB2 utilizzando la rappresentazione WKB.

Durante il trasferimento di un valore dal server dei database al client, per convertire una forma geometrica nella rappresentazione WKT, viene utilizzata la


```

// impostare il gruppo di trasformazione per tutte le istruzioni SQL
EXEC SQL
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_Shape;

// inizializzare le variabili host
...

SET CURRENT DEFAULT TRANSFORM GROUP = ST_Shape;

// inserire forma geometrica utilizzando la rappresentazione shape
nella colonna di tipo ST_Geometry
EXEC SQL
    INSERT
    INTO transforms_sample(id, geom)
    VALUES ( :id, :shape_buffer );

```

Gruppo di trasformazione ST_GML

Il gruppo di trasformazione ST_GML può essere utilizzato per trasmettere i dati da e a DB2[®] utilizzando il linguaggio GML (geography markup language).

Durante il trasferimento di un valore dal server dei database al client, per convertire una forma geometrica nella rispettiva rappresentazione GML, viene utilizzata la stessa funzione fornita da ST_AsGML(). Durante il trasferimento della rappresentazione GML di una forma geometrica al server dei database, viene utilizzata implicitamente la funzione ST_Geometry(CLOB) per eseguire le conversioni in un valore ST_Geometry. Utilizzando il gruppo di trasformazione per collegare i valori a DB2, le forme geometriche verranno rappresentate nel sistema di riferimento spaziale con identificativo 0 (zero).

Esempi:

Negli esempi seguenti, le righe dei risultati sono state formattate nuovamente in un formato leggibile. La spaziatura nei risultati varia in base al tipo di schermo utilizzato.

Esempio 1:

Il seguente script SQL mostra come utilizzare il gruppo di trasformazione ST_GML per richiamare una forma geometrica nella rispettiva rappresentazione GML senza servirsi della funzione ST_AsGML esplicita.

```

CREATE TABLE transforms_sample (
    id INTEGER,
    geom db2gse.ST_Geometry)

INSERT
    INTO transforms_sample
    VALUES ( 1, db2gse.ST_Geometry('multipoint z (10 10
        3, 20 20 4, 15 20 30)', 0) )
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_GML

SELECT id, geom
FROM transforms_sample
WHERE id = 1

```

Risultati:

```

ID      GEOM
-----
1 <gml:MultiPoint srsName=UNSPECIFIED><gml:PointMember>
  <gml:Point><gml:coord><gml:X>10</gml:X>

```

ST_GML

```
<gml:Y>10</gml:Y><gml:Z>3</gml:Z>
</gml:coord></gml:Point></gml:PointMember>
<gml:PointMember><gml:Point><gml:coord>
<gml:X>20</gml:X><gml:Y>20</gml:Y>
<gml:Z>4</gml:Z></gml:coord></gml:Point>
</gml:PointMember><gml:PointMember><gml:Point>
<gml:coord><gml:X>15</gml:X><gml:Y>20
</gml:Y><gml:Z>30</gml:Z></gml:coord>
</gml:Point></gml:PointMember></gml:MultiPoint>
```

Esempio 2:

Il seguente codice C mostra come utilizzare il gruppo di trasformazione ST_GML per inserire le forme geometriche senza servirsi della funzione ST_Geometry esplicita per la variabile host gml_buffer, di tipo CLOB, che contiene la rappresentazione GML del punto (20 ,20) da inserire.

```
EXEC SQL BEGIN DECLARE SECTION;
    sqlint32 id = 0;
    SQL TYPE IS db2gse.ST_Geometry AS CLOB(1000) gml_buffer;
EXEC SQL END DECLARE SECTION;

// impostare il gruppo di trasformazione per tutte le istruzioni SQL
EXEC SQL
    SET CURRENT DEFAULT TRANSFORM GROUP = ST_GML;
    id = 100;
strcpy(gml_buffer.data, "<gml:point><gml:coord>"
    "<gml:X>20</gml:X> <gml:Y>20</gml:Y></gml:coord></gml:point>");

// inizializzare le variabili host
wkt_buffer.length = strlen(gml_buffer.data);

// inserire un punto utilizzando WKT nella colonna di tipo ST_Geometry
EXEC SQL
    INSERT
    INTO transforms_sample(id, geom)
    VALUES ( :id, :gml_buffer );
```

Capitolo 25. Formati dati supportati

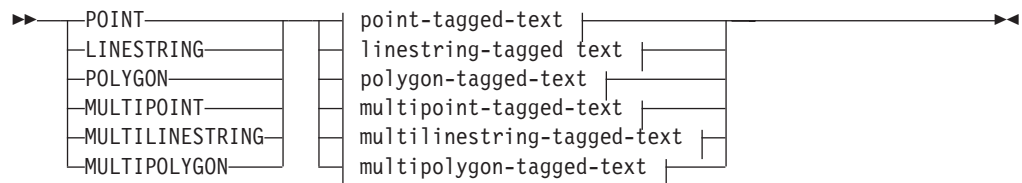
In questo capitolo vengono descritti i formati dei dati spaziali standard industriali che possono essere utilizzati con DB2 Spatial Extender. Per informazioni sulle funzioni che accettano e producono questo tipo di formati, consultare la sezione "Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati" a pagina 295. Per informazioni sull'importazione e l'esportazione dei file contenenti questi formati, consultare la sezione "Informazioni sull'importazione e l'esportazione dei dati spaziali" a pagina 87. Nel capitolo vengono descritti i quattro seguenti formati di dati spaziali:

- Rappresentazione WKT (Well-known text)
- Rappresentazione WKB (Well-known binary)
- Rappresentazione shape
- Rappresentazione GML (Geography Markup Language)

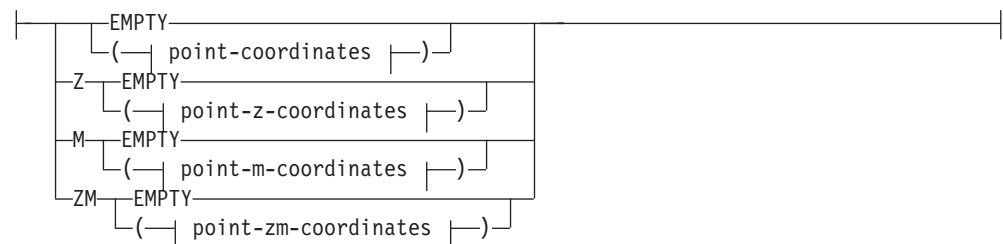
Rappresentazione WKT (well-known text)

La specifica OpenGIS Consortium "Simple Features for SQL" definisce la rappresentazione WKT per lo scambio dei dati geometrici in formato ASCII. La rappresentazione è anche citata nello standard ISO "SQL/MM Part: 3 Spatial". Per ulteriori informazioni sulle funzioni che accettano e producono dati WKT, consultare la sezione "Funzioni spaziali che convertono le forme geometriche in e da formati di scambio dati".

La rappresentazione WKT di una forma geometrica, viene definita come segue:

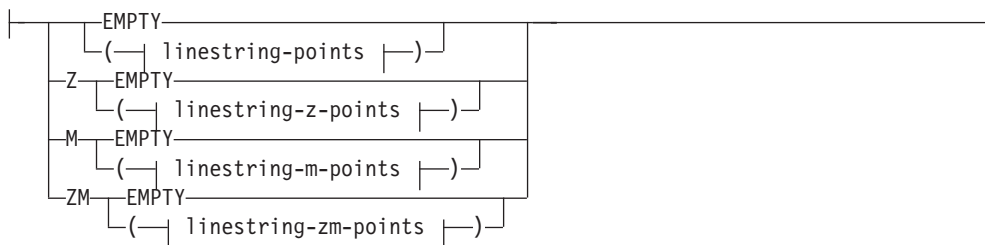


point-tagged-text:

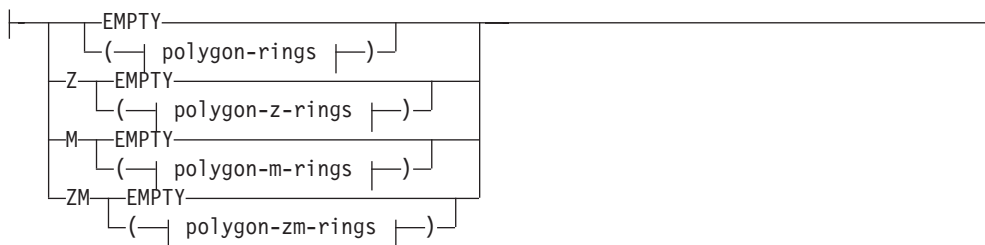


linestring-tagged-text:

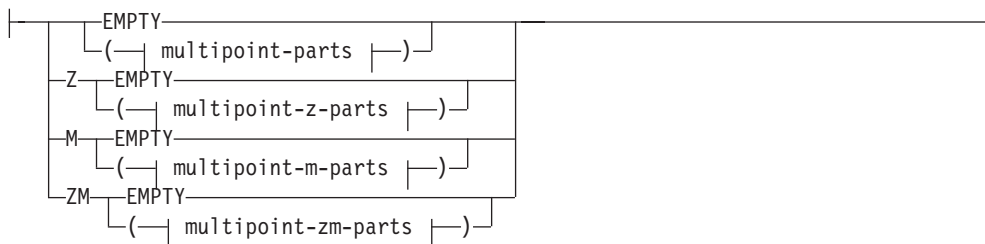
Rappresentazione WKT (Well-known text)



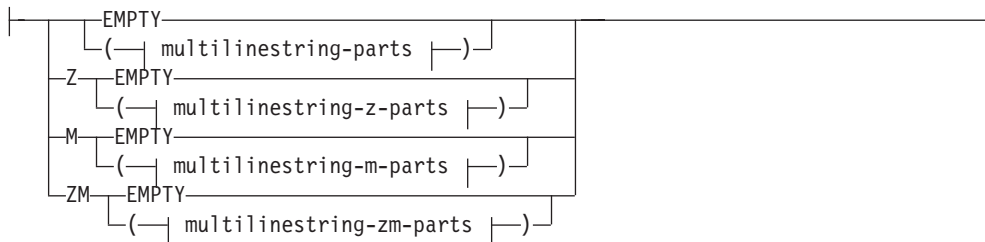
polygon-tagged-text:



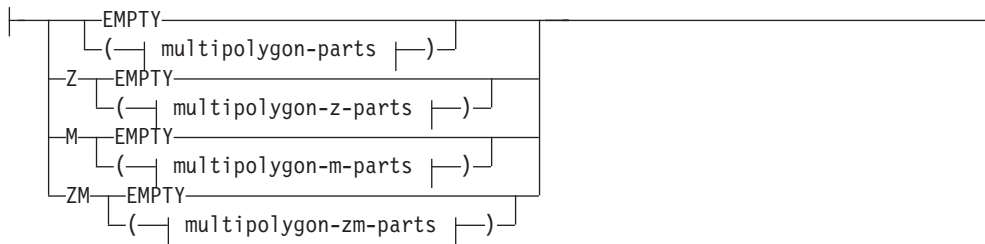
multipoint-tagged-text:



multilinestring-tagged-text:



multipolygon-tagged-text:



point-coordinates:



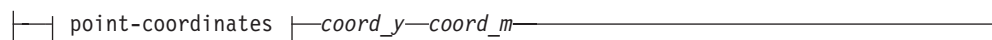
point-z-coordinates:



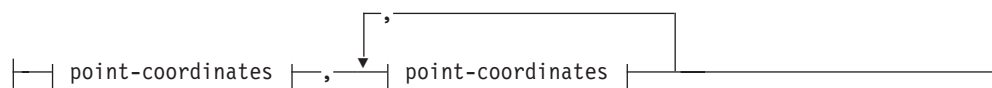
point-m-coordinates:



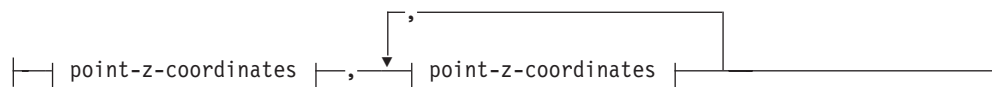
point-zm-coordinates:



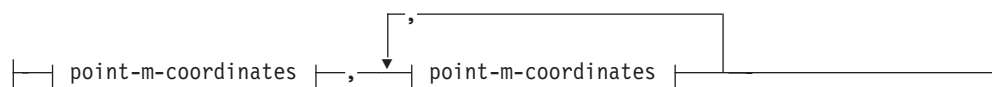
linestring-points:



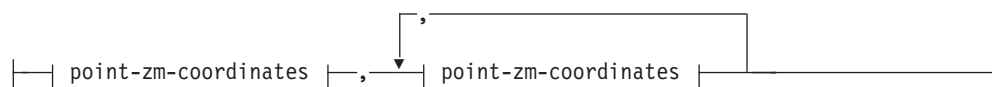
linestring-z-points:



linestring-m-points:



linestring-zm-points:

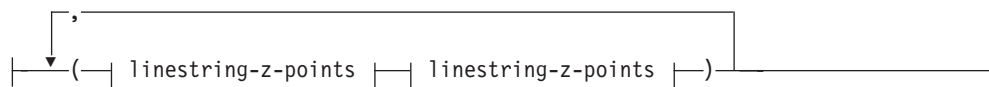


polygon-rings:



Rappresentazione WKT (Well-known text)

polygon-z-rings:



polygon-m-rings:



polygon-zm-rings:



multipoint-parts:



multipoint-z-parts:



multipoint-m-parts:



multipoint-zm-parts:



multilinestring-parts:



multilinestring-z-parts:



multilinestring-m-parts:



multilinestring-zm-parts:



multipolygon-parts:



multipolygon-z-parts:



multipolygon-m-parts:



multipolygon-zm-parts:



Parametri:

coord_x

Un valore numerico (fisso, intero o a virgola mobile), che rappresenta la coordinata X di un punto.

coord_y

Un valore numerico (fisso, intero o a virgola mobile), che rappresenta la coordinata Y di un punto.

Rappresentazione WKT (Well-known text)

coord_z

Un valore numerico (fisso, intero o a virgola mobile), che rappresenta la coordinata Z di un punto.

coord_m

Un valore numerico (fisso, intero o a virgola mobile), che rappresenta la coordinata M (misura) di un punto.

Se la forma geometrica è vuota, invece dell'elenco di coordinate sarà necessario specificare la parola chiave EMPTY. La parola chiave EMPTY non deve essere inserita insieme all'elenco di coordinate.

La seguente tabella fornisce alcuni esempio di possibili rappresentazioni di testo.

Tabella 57. Tipi di forme geometriche e relative rappresentazioni

Tipo di forma geometrica	Rappresentazione WKT	Commento
punto	POINT EMPTY	punto vuoto
punto	POINT (10,05 10,28)	punto
punto	POINT Z(10,05 10,28 2,51)	punto con coordinate Z
punto	POINT M(10,05 10,28 4,72)	punto con coordinate M
punto	POINT ZM(10,05 10,28 2,51 4,72)	punto con coordinate Z e coordinate M
linea	LINestring EMPTY	linea vuota
poligono	POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))	poligono
multipunto	MULTIPOINT Z(10 10 2, 20 20 3)	multipunto con coordinate Z
multilinea	MULTILINESTRING M(((310 30 1, 40 30 20, 50 20 10)(10 10 0, 20 20 1))	multilinea con coordinate M
multipartigono	MULTIPOLYGON ZM(((1 1 1 1, 1 2 3 4, 2 2 5 6, 2 1 7 8, 1 1 1 1)))	multipartigono con coordinate Z ed M

Riferimenti correlati:

- "Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati" a pagina 295

Rappresentazione WKB (well-known binary)

Questa sezione descrive la rappresentazione WKB per le forme geometriche.

La specifica OpenGIS Consortium "Simple Features for SQL" definisce la rappresentazione WKB. La rappresentazione è anche definita nello standard International Organization for Standardization (ISO) "SQL/MM Part: 3 Spatial". Per ulteriori informazioni sulle funzioni che accettano e producono la WKB, vedere i riferimenti correlati riportati al termine di questa sezione.

Rappresentazione WKB (Well-known binary)

Il punto di partenza per le rappresentazioni WKB consiste nel flusso di dati di un punto, che consiste in due valori doppi. I flussi di dati delle altre forme geometriche sono creati utilizzando i flussi delle forme già definite.

L'esempio seguente mostra il blocco di creazione di base per le rappresentazioni WKB.

```
// Basic Type definitions
// byte : 1 byte
// uint32 : 32 bit unsigned integer (4 bytes)
// double : double precision number (8 bytes)

// Building Blocks : Point, LinearRing

Point {
    double x;
    double y;
};
LinearRing {
    uint32 numPoints;
    Point points[numPoints];
};
enum wkbGeometryType {
    wkbPoint = 1,
    wkbLineString = 2,
    wkbPolygon = 3,
    wkbMultiPoint = 4,
    wkbMultiLineString = 5,
    wkbMultiPolygon = 6
};
enum wkbByteOrder {
    wkbXDR = 0, // Big Endian
    wkbNDR = 1 // Little Endian
};
WKBPoint {
    byte byteOrder;
    uint32 wkbType; // 1=wkbPoint
    Point point;
};
WKBLineString {
    byte byteOrder;
    uint32 wkbType; // 2=wkbLineString
    uint32 numPoints;
    Point points[numPoints];
};
WKBPolygon {
    byte byteOrder;
    uint32 wkbType; // 3=wkbPolygon
    uint32 numRings;
    LinearRing rings[numRings];
};
WKBMultiPoint {
    byte byteOrder;
    uint32 wkbType; // 4=wkbMultipoint
    uint32 num_wkbPoints;
    WKBPoint wkbPoints[num_wkbPoints];
};
WKBMultiLineString {
    byte byteOrder;
    uint32 wkbType; // 5=wkbMultiLineString
    uint32 num_wkbLineStrings;
    WKBLineString wkbLineStrings[num_wkbLineStrings];
};
wkbMultiPolygon {
    byte byteOrder;
```

Rappresentazione WKB (Well-known binary)

```
uint32      wkbType;    // 6=wkbMultiPolygon
uint32      num_wkbPolygons;
WKBPolygon  wkbPolygons[num_wkbPolygons];
};

WKBGeometry {
  union {
    WKBPoint      point;
    WKBLineString linestring;
    WKBPolygon     polygon;
    WKBMultiPoint mpoint;
    WKBMultiLineString mlinestring;
    WKBMultiPolygon mpolygon;
  }
};
```

La figura seguente mostra un esempio di forma geometrica in una rappresentazione WKB, utilizzando il codice NDR.

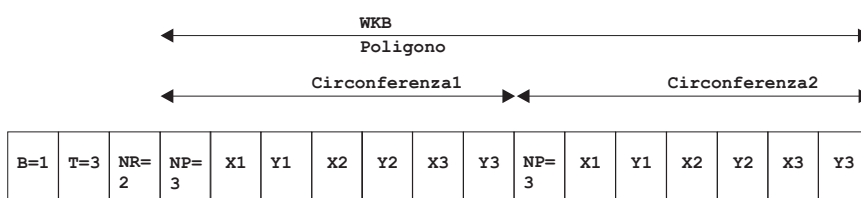


Figura 58. Rappresentazione della forma geometrica in formato NDR. (B=1) del poligono di tipo (T=3) con 2 (NR=2) lineari, in cui ciascuna circonferenza ha 3 punti (NP=3).

Riferimenti correlati:

- “Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati” a pagina 295

Rappresentazione shape

Le rappresentazioni shape sono ampiamente utilizzate negli standard industriali definiti da ESRI. Per una descrizione completa delle rappresentazioni shape, fare riferimento al sito Web di ESRI all'indirizzo <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

La sezione “Funzioni spaziali per la conversione di forme geometriche in e da formati di scambio dati” seguente, descrive le funzioni spaziali che accettano e producono il formato di dati shape.

Riferimenti correlati:

- “Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati” a pagina 295

Rappresentazione GML (Geography Markup Language)

DB2 Spatial Extender dispone di diverse funzioni che generano forme geometriche da rappresentazioni in GML (geography markup language).

Per una descrizione dettagliata delle funzioni fornite da DB2 Spatial Extender per la conversione dei valori di forma geometrica in e da rappresentazioni GML, consultare la sezione “Funzioni spaziali per la conversione di forme geometriche in e da formati di scambio dati” segnalata di seguito.

Il linguaggio GML (Geography Markup Language) è una codifica XML per le informazioni geografiche definite dalla specifica dell'OpenGIS Consortium "Geography Markup Language V2". Tale specifica è riportata all'indirizzo <http://www.opengis.org/techno/implementation.htm>.

Riferimenti correlati:

- "Funzioni spaziali che convertono i valori delle forme geometriche in formati di scambio dati" a pagina 295

Rappresentazione GML

Capitolo 26. Sistemi di coordinate supportati

Questo capitolo fornisce informazioni di riferimento sui valori di coordinate utilizzati per interpretare i dati spaziali. Vengono trattati i seguenti argomenti:

- Panoramica sui sistemi di coordinate
- Unità di misura lineari supportate
- Unità di misura angolari supportate
- Sferoidi supportati
- Datum geodetici supportati
- Primi meridiani supportati
- Proiezioni di carte geografiche supportate

Sistemi di coordinate supportati

Questa sezione contiene una descrizione della sintassi dei sistemi di coordinate e un elenco dei valori dei sistemi di coordinate supportati da DB2 Spatial Extender.

Sintassi dei sistemi di coordinate

La rappresentazione WKT dei sistemi di riferimento spaziali fornisce una rappresentazione di testo standard delle informazioni dei sistemi di coordinate. Le definizioni della rappresentazione WKT sono definite dalla specifica OGC "Simple Features for SQL" e dallo standard ISO SQL/MM Part 3: Spatial standard.

Un sistema di coordinate può essere geografico (latitudine-longitudine), proiettato (X,Y), o geocentrico (X,Y,Z). Il sistema di coordinate è composto da diversi oggetti. Ciascun oggetto dispone di una parola chiave in caratteri maiuscoli (ad esempio DATUM o UNIT) seguita da parametri di definizione, separati da virgola, dell'oggetto, in parentesi. Alcuni oggetti sono composti da altri oggetti, quindi il risultato sarà una struttura nidificata.

Nota: Nelle implementazioni è possibile sostituire le parentesi tonde () con parentesi quadrate [].

La definizione EBNF (Extended Backus Naur Form) per la rappresentazione in stringhe di un sistema di coordinate che utilizza parentesi quadrate è la seguente (fare riferimento alla nota precedente per l'uso delle parentesi):

```
<coordinate system> = <projected cs> |  
<geographic cs> | <geocentric cs>  
<projected cs> = PROJCS["<name>",  
<geographic cs>, <projection>, {<parameter>,*  
<linear unit>}]  
<projection> = PROJECTION["<name>"]  
<parameter> = PARAMETER["<name>",  
<value>]  
  
<value> = <number>
```

Il tipo di sistema di coordinate viene identificato dalla parola chiave utilizzata:

PROJCS

Un sistema di coordinate viene identificato dalla parola chiave PROJCS se i dati appartengono a coordinate proiettate

Sistemi di coordinate supportati

GEOGCS

Un sistema di coordinate viene identificato dalla parola chiave GEOGCS se i dati appartengono a coordinate geografiche

GEOCCS

Un sistema di coordinate viene identificato dalla parola chiave GEOCCS se i dati appartengono a coordinate geocentriche

La parola chiave PROJCS è seguita da tutti i "componenti" che definiscono il sistema di coordinate proiettate. Il primo componente è sempre il nome. Il nome del sistema di coordinate proiettate è seguito da diversi oggetti: il sistema di coordinate geografiche, la proiezione della mappa, uno o più parametri e l'unità di misura lineare. Tutti i sistemi di coordinate proiettate si basano su un sistema di coordinate geografiche, quindi di seguito verranno descritti innanzitutto i componenti esclusivi dei sistemi di coordinate proiettate. Ad esempio, viene definita la zona UTM 10N nel datum NAD83:

```
PROJCS["NAD_1983_UTM_Zone_10N",  
<geographic cs>,  
PROJECTION["Transverse_Mercator"],  
PARAMETER["False_Easting",500000.0],  
PARAMETER["False_Northing",0.0],  
PARAMETER["Central_Meridian",-123.0],  
PARAMETER["Scale_Factor",0.9996],  
PARAMETER["Latitude_of_Origin",0.0],  
UNIT["Meter",1.0]]
```

Il sistema di coordinate geografiche viene definito dal nome e da una serie di oggetti in quest'ordine: il datum, il primo meridiano e l'unità di misura angolare.

```
<geographic cs> = GEOGCS["<name>", <datum>, <prime meridian>, <angular unit>]  
<datum> = DATUM["<name>", <spheroid>]  
<spheroid> = SPHEROID["<name>", <semi-major axis>, <inverse flattening>]  
<semi-major axis> = <number>  
<inverse flattening> = <number>  
<prime meridian> = PRIMEM["<name>", <longitude>]  
<longitude> = <number>
```

L'asse semi maggiore deve essere misurato in metri e deve essere maggiore di zero.

Stringa del sistema di coordinate geografiche per la zona UTM 10 su NAD83:

```
GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",  
SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],  
UNIT["Degree",0.0174532925199433]]
```

L'oggetto UNIT può rappresentare unità di misura angolari o lineari:

```
<angular unit> = <unit>  
<linear unit> = <unit>  
<unit> = UNIT["<name>", <conversion factor>]  
<conversion factor> = <number>
```

Il fattore di conversione specifica il numero di metri (unità lineare) o il numero di radianti (unità angolare) per unità, che dovrà essere maggiore di zero.

La rappresentazione in stringa completa della zona UTM 10N sarà la seguente:

```
PROJCS["NAD_1983_UTM_Zone_10N",  
GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433]],
```



```
PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],
PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-123.0],
PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_of_Origin",0.0],
UNIT["Meter",1.0]
```

Il sistema di coordinate geocentrico è molto simile al sistema di coordinate geografiche:

```
<geocentric cs> = GEOCCS["<name>", <datum>, <prime meridian>, <linear unit>]
```

Unità di misura lineari supportate

Tabella 58. Unità di misura lineari supportate

Unità di misura	Fattore di conversione
Metri	1,0
Piede (Internazionale)	0,3048
Piede U.S.	12/39,37
Modified American Foot	12,0004584/39,37
Clarke's Foot	12/39,370432
Indian Foot	12/39,370141
Link	7,92/39,370432
Link (Benoit)	7,92/39,370113
Link (Sears)	7,92/39,370147
Chain (Benoit)	792/39,370113
Chain (Sears)	792/39,370147
Yard (Indian)	36/39,370141
Yard (Sears)	36/39,370147
Braccio	1,8288
Miglio marino	1852,0

Unità di misura angolari supportate

Tabella 59. Unità di misura angolari supportate

Unità di misura	Intervallo valido per la latitudine	Intervallo valido per la longitudine	Fattore di conversione
Radiante	radianti $-\pi/2$ e $\pi/2$ (inclusivo)	radianti $-\pi$ e π (inclusivo)	1,0
Grado decimale	gradi -90 e 90 (inclusivo)	gradi -180 e 180 (inclusivo)	$\pi/180$
Minuto decimale	minuti -5400 e 5400 (inclusivo)	minuti -10800 e 10800 (inclusivo)	$(\pi/180)/60$
Secondo decimale	secondi -324000 e 324000 (inclusivo)	secondi -648000 e 648000 (inclusivo)	$(\pi/180)*3600$
Gon	gradianti -100 e 100 (inclusivo)	gradianti -200 e 200 (inclusivo)	$\pi/200$
Grado	gradianti -100 e 100 (inclusivo)	gradianti -200 e 200 (inclusivo)	$\pi/200$

Sferoidi supportati

Tabella 60. Sferoidi supportati

Nome	Asse semi maggiore	Schiacciamento inverso
Airy 1830	6377563,396	299,3249646
Airy Modified 1849	6377340,189	299,3249646
Average Terrestrial System 1977	6378135.0	298.257
Australian National Spheroid	6378160.0	298,25
Bessel 1841	6377397,155	299,1528128
Bessel Modified	6377492,018	299,1528128
Bessel Namibia	6377483,865	299,1528128
Clarke 1858	6378293.639	294.260676369
Clarke 1866	6378206,4	294,9786982
Clarke 1866 (Michigan)	6378450.047	294,978684677
Clarke 1880	6378249.138	293,466307656
Clarke 1880 (Arc)	6378249,145	293,466307656
Clarke 1880 (Benoit)	6378300,79	293,466234571
Clarke 1880 (IGN)	6378249,2	293,46602
Clarke 1880 (RGS)	6378249,145	293,465
Clarke 1880 (SGA 1922)	6378249,2	293,46598
Everest (1830 Definition)	6377299.36	300,8017
Everest 1830 Modified	6377304,063	300,8017
Everest Adjustment 1937	6377276,345	300,8017
Everest 1830 (1962 Definition)	6377301,243	300.8017255
Everest 1830 (1967 Definition)	6377298,556	300,8017
Everest 1830 (1975 Definition)	6377299.151	300.8017255
Everest 1969 Modified	6377295.664	300,8017
Fischer 1960	6378166.0	298,3
Fischer 1968	6378150 .0	298,3
Modified Fischer	6378155 .0	298,3
GEM 10C	6378137.0	298,257222101
GRS 1967	6378160.0	298.247167427
GRS 1967 Truncated	6378160.0	298,25
GRS 1980	6378137.0	298,257222101
Helmert 1906	6378200.0	298,3
Hough 1960	6378270.0	297,0
Indonesian National Spheroid	6378160.0	298.247
International 1924	6378388.0	297,0
International 1967	6378160.0	298,25
Krassowsky 1940	6378245.0	298,3
NWL 9D	6378145.0	298,25
NWL 10D	6378135.0	298,26

Tabella 60. Sferoidi supportati (Continua)

Nome	Asse semi maggiore	Schiacciamento inverso
OSU 86F	6378136,2	298,25722
OSU 91A	6378136,3	298,25722
Plessis 1817	6376523.0	308,64
Sphere	6371000.0	0.0
Sphere (ArcInfo)	6370997.0	0.0
Struve 1860	6378298.3	294,73
Walbeck	6376896.0	302,78
War Office	6378300.0	296.0
WGS 1966	6378145.0	298,25
WGS 1972	6378135.0	298,26
WGS 1984	6378137.0	298,257223563

Datum geodetici supportati

Tabella 61. Datum geodetici supportati

Nome	Datum geodetico
Adindan	Lisbon
Afgooye	Loma Quintana
Agadez	Lome
Australian Geodetic Datum 1966	Luzon 1911
Australian Geodetic Datum 1984	Mahe 1971
Ain el Abd 1970	Makassar
Amersfoort	Malongo 1987
Aratu	Manoca
Arc 1950	Massawa
Arc 1960	Merchich
Ancienne Triangulation Francaise	Militar-Geographische Institute
Barbados	Mhast
Batavia	Minna
Beduaram	Monte Mario
Beijing 1954	M'poraloko
Reseau National Belge 1950	NAD Michigan
Reseau National Belge 1972	North American Datum 1927
Bermuda 1957	North American Datum 1983
Bern 1898	Nahrwan 1967
Bern 1938	Naparima 1972
Ancienne Triangulation Francaise	Militar-Geographische Institute
Barbados	Mhast
Batavia	Minna
Beduaram	Monte Mario

Sistemi di coordinate supportati

Tabella 61. Datum geodetici supportati (Continua)

Nome	Datum geodetico
Beijing 1954	M'poraloko
Reseau National Belge 1950	NAD Michigan
Reseau National Belge 1972	North American Datum 1927
Bermuda 1957	North American Datum 1983
Bern 1898	Nahrwan 1967
Bern 1938	Naparima 1972
Ancienne Triangulation Francaise	Militar-Geographische Institute
Barbados	Mhast
Batavia	Minna
Beduaram	Monte Mario
Beijing 1954	M'poraloko
Reseau National Belge 1950	NAD Michigan
Reseau National Belge 1972	North American Datum 1927
Bermuda 1957	North American Datum 1983
Bern 1898	Nahrwan 1967
Bern 1938	Naparima 1972
Bogota	Nord de Guerre
Bukit Rimpah	NGO 1948
Camacupa	Nord Sahara 1959
Campo Inchauspe	NSWC 9Z-2
Cape	Nouvelle Triangulation Francaise
Carthage	New Zealand Geodetic Datum 1949
Chua	OS (SN) 1980
Conakry 1905	OSGB 1936
Corrego Alegre	OSGB 1970 (SN)
Cote d'Ivoire	Padang 1884
Datum 73	Palestine 1923
Deir ez Zor	Pointe Noire
Deutsche Hauptdreiecksnetz	Provisional South American Datum 1956
Douala	Pulkovo 1942
European Datum 1950	Qatar
European Datum 1987	Qatar 1948
Egypt 1907	Qornoq
European Reference System 1989	RT38
Fahud	South American Datum 1969
Gandajika 1970	Sapper Hill 1943
Garoua	Schwarzeck
Geocentric Datum of Australia 1994	Segora
Guyane Francaise	Serindung
Herat North	Stockholm 1938

Tabella 61. Datum geodetici supportati (Continua)

Nome	Datum geodetico
Hito XVIII 1963	Sudan
Hu Tzu	Shan Tananarive 1925
Hungarian Datum 1972	Timbalai 1948
Indian 1954	TM65
Indian 1975	TM75
Indonesian Datum 1974	Tokyo
Jamaica 1875	Trinidad 1903
Jamaica 1969	Trucial Coast 1948
Kalianpur	Voirol 1875
Kandawala	Voirol Unifie 1960
Kertau	WGS 1972
Kuwait Oil Company	WGS 1972 Transit Broadcast Ephemeris
La Canoa	WGS 1984
Lake	Yacare
Leigon	Yoff
Liberia 1964	Zanderij

Primi meridiani supportati

Tabella 62. Primi meridiani supportati

Posizione	Coordinate
Greenwich	0° 0' 0"
Berna	7° 26' 22.5" E
Bogota	74° 4' 51.3" W
Bruxelles	4° 22' 4.71" E
Ferro	17° 40' 0" W
Giacarta	106° 48' 27.79" E
Lisbona	9° 7' 54.862" W
Madrid	3° 41' 16.58" W
Parigi	2° 20' 14.025" E
Roma	12° 27' 8.4" E
Stoccolma	18° 3' 29" E

Proiezioni di mappa supportate

Tabella 63. Proiezioni cilindriche

Proiezioni cilindriche	Proiezioni pseudocilindriche
Behrmann	Craster parabolic
Cassini	Eckert I
Cylindrical equal area	Eckert II
Equirectangular	Eckert III

Sistemi di coordinate supportati

Tabella 63. Proiezioni cilindriche (Continua)

Proiezioni cilindriche	Proiezioni pseudocilindriche
Gall's stereographic	Eckert IV
Gauss-Kruger	Eckert V
Mercator	Eckert VI
Miller cylindrical	McBryde-Thomas flat polar quartic
Oblique	Mercator (Hotine) Mollweide
Plate-Carée	Robinson
Times	Sinusoidal (Sansom-Flamsteed)
Transverse Mercator	Winkel I

Tabella 64. Proiezioni coniche

Nome	Proiezione conica
Albers conic equal-area	Chamberlin trimetric
Bipolar oblique conformal conic	Two-point equidistant
Bonne	Hammer-Aitoff equal-area
Equidistant conic	Van der Grinten I
Lambert conformal conic	Miscellaneous
Polyconic	Alaska series E
Simple conic	Alaska Grid (Modified-Stereographic by Snyder)

Tabella 65. Parametri di proiezione mappa

Parametro	Descrizione
central_meridian	La linea di longitudine scelta come origine delle coordinate x.
scale_factor	Scale_factor viene utilizzato generalmente per ridurre la quantità di distorsione in una proiezione di mappa.
standard_parallel_1	Una linea di latitudine che generalmente non ha distorsione. Utilizzata anche per la latitudine della scala reale.
standard_parallel_2	Una linea di longitudine che generalmente non ha distorsione.
longitude_of_center	La longitudine che definisce il punto centrale della proiezione della mappa.
nlatitude_of_center	La latitudine che definisce il punto centrale della proiezione della mappa.
longitude_of_origin	La longitudine scelta come origine delle coordinate x.
latitude_of_origin	La latitudine scelta come origine delle coordinate y.
false_easting	Un valore aggiunto alle coordinate x in modo che risultino tutte positive.
false_northing	Un valore aggiunto alle coordinate y in modo che risultino tutte positive.

Tabella 65. Parametri di proiezione mappa (Continua)

Parametro	Descrizione
azimuth	L'angolo che definisce la linea centrale di una proiezione obliqua.
longitude_of_point_1	La longitudine del primo punto necessario per una proiezione di mappa.
latitude_of_point_1	La latitudine del primo punto necessario per una proiezione di mappa.
longitude_of_point_2	La longitudine del secondo punto necessario per una proiezione di mappa.
latitude_of_point_2	La latitudine del secondo punto necessario per una proiezione di mappa.
longitude_of_point_3	La longitudine del terzo punto necessario per una proiezione di mappa.
latitude_of_point_3	La latitudine del terzo punto necessario per una proiezione di mappa.
landsat_number	Il numero del satellite Landsat.
path_number	Il numero del percorso orbitale di un determinato satellite.
perspective_point_height	L'altezza dalla Terra del punto di prospettiva della proiezione della mappa.
fipszone	Numero di zona dello State Plane Coordinate System.
zone	Numero di zona UTM.

Materiale di riferimento

Appendice A. Procedure memorizzate obsolete

In questa sezione vengono descritte le procedure memorizzate obsolete.

Nota: Suggerimento: scrivere tutte le nuove applicazioni utilizzando le procedure memorizzate definite in DB2 Spatial Extender Versione 8 ed aggiornare le applicazioni correnti in modo che utilizzino le procedure definite nella Versione 8.

Le procedure memorizzate obsolete eseguono le attività descritte nella seguente tabella.

Tabella 66. Procedure memorizzate obsolete

Nome della procedura memorizzata	Attività della procedura memorizzata
db2gse.gse_enable_autogc	Abilita il geocoder per la sincronizzazione automatica delle colonne spaziali con le colonne di attributi corrispondenti
db2gse.gse_enable_db	Abilita il database per il supporto delle operazioni spaziali
db2gse.gse_enable_idx	Crea indici per le colonne spaziali
db2gse.gse_enable_sref	Crea un sistema di riferimento spaziale
db2gse.gse_export_shape	Esporta un livello e la tabella associata in un file shape
db2gse.gse_disable_autogc	Disabilita il geocoder in modo che non potesse sincronizzare automaticamente le colonne spaziali con le colonne di attributi corrispondenti
db2gse.gse_disable_db	Disabilita il supporto per le operazioni spaziali nei database
db2gse.gse_disable_sref	Elimina i sistemi di riferimento spaziali
db2gse.gse_import_shape	Importa un livello e la tabella associata da un file di trasferimento ESRI_SDE
db2gse.gse_register_gc	Registra un geocoder diverso dal geocoder predefinito
db2gse.gse_register_layer	Registra una colonna spaziale come livello
db2gse.gse_run_gc	Esegue il geocoder in modalità batch
db2gse.gse_unregist_gc	Annula la registrazione dei geocoder diversi dal geocoder predefinito
db2gse.gse_unregist_layer	Annula la registrazione di un livello

db2gse.gse_enable_autogc

Utilizzare questa procedura memorizzata per:

- Creare trigger che sincronizzino la colonna spaziale con le corrispondenti colonne di attributi. Ogni qual volta vengono inseriti o aggiornati gli attributi nelle colonne di attributi, il trigger richiama un geocoder registrato che esegue il geocoding dei nuovi valori e inserisce i risultati nella colonna spaziale.
- Riattivare i trigger dopo una disabilitazione temporanea.

Procedure memorizzate obsolete

- Stabilire le funzioni da utilizzare per eseguire il geocoding dei valori inseriti o aggiornati.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella in cui sono definiti i trigger creati da questa procedura memorizzata.
- Privilegio CONTROL per la tabella.
- Privilegi ALTER, SELECT e UPDATE per la tabella.

Parametri:

Tabella 67. Parametri di input per la procedura memorizzata `db2gse.gse_enable_autogc`.

Nome	Tipo di dati	Descrizione
<code>modo_oper</code>	SMALLINT	<p>Valore che indica se i trigger che avviano il geocoding devono essere creati per la prima volta o devono essere riattivati dopo una disabilitazione temporanea.</p> <p>Il parametro non può essere nullo.</p> <p>Commento: Per creare i trigger, utilizzare la macro <code>GSE_AUTOGC_CREATE</code>. Per riattivarli, utilizzare la macro <code>GSE_AUTOGC_RECREATE</code>. Per individuare i valori associati alle macro, fare riferimento al file <code>db2gse.h</code>. In ambiente AIX, il file è memorizzato nella directory <code>\$DB2INSTANCE/sqlib/include/</code>. In ambiente Windows NT, è memorizzato nella directory <code>%DB2PATH%\include\</code>.</p> <p>Se il parametro <code>modo_oper</code> è impostato su <code>GSE_AUTOGC_CREATE</code>, è necessario assegnare un identificativo di un geocoder registrato al parametro <code>gcId</code>.</p>
<code>schemalivello</code>	VARCHAR(30)	<p>Nome dello schema a cui appartiene la tabella specificata nel parametro <code>tabellalivello</code>.</p> <p>Il parametro può essere nullo.</p> <p>Se non viene indicato alcun valore per il parametro <code>schemalivello</code>, per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata <code>db2gse.gse_enable_autogc</code>.</p>
<code>tabellalivello</code>	VARCHAR(128)	<p>Nome della tabella che verrà utilizzata dai trigger creati o riattivati da questa procedura memorizzata.</p> <p>Il parametro non può essere nullo.</p>
<code>colonnalivello</code>	VARCHAR(128)	<p>Nome della colonna spaziale che verrà gestita dai trigger creati o riattivati da questa procedura memorizzata.</p> <p>Il parametro non può essere nullo.</p> <p>Il parametro <code>colonnalivello</code> deve fare riferimento ad una colonna registrata come livello di tabella.</p>

Tabella 67. Parametri di input per la procedura memorizzata *db2gse.gse_enable_autogc*. (Continua)

Nome	Tipo di dati	Descrizione
IDgc	INTEGER	Identificativo del geocoder che verrà richiamato dai trigger di inserimento e aggiornamento creati o riattivati da questa procedura memorizzata. Il parametro non può essere nullo, il parametro <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_CREATE</i> . Sarà nullo se <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_RECREATE</i> .
livello_precision	INTEGER	Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento, affinché il geocoder elabori i dati di origine correttamente. Il parametro non può essere nullo se il parametro <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_CREATE</i> . Sarà nullo se <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_RECREATE</i> . Il livello di precisione può variare dall'1 al 100 per cento.
specifiche_fornitore	VARCHAR(256)	Informazioni tecniche fornite dal fornitore; ad esempio, il nome e il percorso di un file utilizzato dal fornitore per impostare i parametri. Il parametro non può essere nullo, il parametro <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_CREATE</i> . Sarà nullo se <i>modo_oper</i> è impostato su <i>GSE_AUTOGC_RECREATE</i> .

Risultati:

Tabella 68. Parametri di output per la procedura memorizzata *db2gse.gse_enable_autogc*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server.

db2gse.gse_enable_db

Utilizzare questa procedura memorizzata per fornire al database risorse necessarie per memorizzare dati spaziali e supportare le operazioni. Queste risorse includono tipi di dati un tipo di indice spaziale, viste e tabelle del catalogo, funzioni e procedure memorizzate. La libreria esterna e il nome della funzione per la procedura memorizzata è *db2gse.gse_enable_db*.

Autorizzazione:

L'ID utente con il quale la procedura memorizzata viene richiamata deve disporre dell'autorizzazione *SYSADM* o *DBADM* per il database che si desidera abilitare.

Procedure memorizzate obsolete

Risultati:

Tabella 69. Parametri di output per la procedura memorizzata *db2gse.gse_enable_db*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_enable_idx

Utilizzare questa procedura memorizzata per creare un indice per una colonna spaziale.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella per la quale verrà utilizzato l'indice abilitato.
- Privilegio CONTROL o INDEX per la tabella.

Parametri:

Tabella 70. Parametri di input per la procedura memorizzata *db2gse.gse_enable_idx*.

Nome	Tipo di dati	Descrizione
schemalivello	VARCHAR(30)	Nome dello schema a cui appartiene la tabella specificata nel parametro <i>tabellalivello</i> . Il parametro può essere nullo. E' necessario indicare un valore per il parametro. Il parametro può avere valore nullo.
tabellalivello	VARCHAR(128)	Nome della tabella per la quale viene definito l'indice creato. Il parametro non può essere nullo.
colonnalivello	VARCHAR(128)	Nome della colonna abilitata per le operazioni spaziali in cui eseguire la ricerca con l'aiuto dell'indice creato. Il parametro non può essere nullo.
nomeindice	VARCHAR(128)	Nome dell'indice da creare. Il parametro non può essere nullo. Non specificare un nome per lo schema. DB2 Spatial Extender assegna automaticamente l'indice allo schema in base al parametro <i>schemalivello</i> .
dimensgriglia_1	DOUBLE	Numero che indica la granulosità della griglia di indice più precisa. Il parametro non può essere nullo.

Tabella 70. Parametri di input per la procedura memorizzata *db2gse.gse_enable_idx*. (Continua)

Nome	Tipo di dati	Descrizione
dimens_griglia_2	DOUBLE	<p>Numero che indica (1) che l'indice non ha una seconda griglia o (2) il livello di granulosità della seconda griglia.</p> <p>Il parametro può essere nullo.</p> <p>Se non esiste una seconda griglia, indicare 0. Se si desidera utilizzare una seconda griglia, il livello di granulosità dovrà essere inferiore al livello indicato dal parametro <i>dimens_griglia_1</i>.</p>
dimens_griglia_3	DOUBLE	<p>Numero che indica (1) che l'indice non ha una terza griglia o (2) il livello di granulosità della terza griglia.</p> <p>Il parametro può essere nullo.</p> <p>Se non esiste una terza griglia, indicare 0. Se si desidera utilizzare una terza griglia, il livello di granulosità dovrà essere inferiore al livello indicato dal parametro <i>dimens_griglia_2</i>.</p>

Risultati:

Tabella 71. Parametri di output per la procedura memorizzata *db2gse.gse_enable_idx*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_enable_sref

Utilizzare questa procedura guidata per indicare la modalità di conversione dei numeri negativi e decimali in interi positivi in un determinato sistema di coordinate, in modo che possano essere memorizzati da DB2 Spatial Extender. L'insieme delle impostazioni effettuate viene definito *sistema di riferimento spaziale*. Durante l'elaborazione della procedura memorizzata, le informazioni relative al sistema di riferimento spaziale vengono aggiunte alla vista del catalogo DB2GSE.SPATIAL_REF_SYS.

Autorizzazione:

Nessuna.

Procedure memorizzate obsolete

Parametri:

Tabella 72. Parametri di input per la procedura memorizzata *db2gse.gse_enable_sref*.

Nome	Tipo di dati	Descrizione
IDsr	INTEGER	Un identificativo numerico per il sistema di riferimento spaziale. L'identificativo deve essere unico nel database abilitato per le operazioni spaziali. Il parametro non può essere nullo.
nome_sr	VARCHAR(64)	Descrizione breve del sistema di riferimento spaziale. La descrizione deve essere unica nel database abilitato per le operazioni spaziali. Il parametro non può essere nullo.
x_falso	DOUBLE	Un numero che, sottratto da un valore di coordinata X negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero). Il parametro non può essere nullo.
y_falso	DOUBLE	Un numero che, sottratto da un valore di coordinata Y negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero). Il parametro non può essere nullo.
unità_xy	DOUBLE	Un numero che, moltiplicato per una coordinata X decimale o per una coordinata Y decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit. Il parametro non può essere nullo.
z_falso	DOUBLE	Un numero che, sottratto da un valore di coordinata Z negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero). Il parametro non può essere nullo.
unità_z	DOUBLE	Un numero che, moltiplicato per una coordinata Z decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit. Il parametro non può essere nullo.
m_falso	DOUBLE	Un numero che, sottratto da una misura negativa, dia come risultato un numero non negativo (ovvero un numero positivo o zero). Il parametro non può essere nullo.
unità_m	DOUBLE	Un numero che, moltiplicato per una misura decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit. Il parametro non può essere nullo.

Tabella 72. Parametri di input per la procedura memorizzata *db2gse.gse_enable_sref*. (Continua)

Nome	Tipo di dati	Descrizione
IDsc	INTEGER	Identificativo numerico del sistema di coordinate da cui deriva il sistema di riferimento spaziale. Per individuare l'identificativo numerico di un sistema di coordinate, fare riferimento alla vista del catalogo DB2GSE.COORD_REF_SYS. Il parametro non può essere nullo.

Risultati:

Tabella 73. Parametri di output per la procedura memorizzata *db2gse.gse_enable_sref*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_export_shape

Utilizzare questa procedura memorizzata per esportare un livello e la tabella associata in un file shape o per creare un nuovo file shape ed esportare un livello e la tabella associata nel nuovo file.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre del privilegio SELECT per la tabella che si desidera esportare.

Parametri:

Tabella 74. Parametri di input per la procedura memorizzata *db2gse.gse_export_shape*.

Nome	Tipo di dati	Descrizione
schemalivello	VARCHAR(30)	Nome dello schema a cui appartiene la tabella specificata nel parametro <i>tabellalivello</i> . Il parametro può essere nullo. Se non viene indicato alcun valore per il parametro <i>schemalivello</i> , per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata <i>db2gse.gse_export_shape</i> .
tabellalivello	VARCHAR(128)	Nome della tabella da esportare. Il parametro non può essere nullo.
colonnalivello	VARCHAR(30)	Nome della colonna che è stata registrata come il livello da esportare. Il parametro non può essere nullo.

Procedure memorizzate obsolete

Tabella 74. Parametri di input per la procedura memorizzata *db2gse.gse_export_shape*. (Continua)

Nome	Tipo di dati	Descrizione
nome_file	VARCHAR(128)	Il nome completo del file shape in cui esportare il livello specificato. Il parametro non può essere nullo.
clausola_where	VARCHAR(1024)	Il testo della clausola where. Definisce una limitazione per la serie di righe da esportare. La clausola può fare riferimento a qualsiasi colonna di attributo nella tabella che si desidera esportare. La parola chiave WHERE non è necessaria in questa clausola. Il parametro può essere nullo.

Risultati:

Tabella 75. Parametri di output per la procedura memorizzata *db2gse.gse_export_shape*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

Limitazione:

E' possibile esportare solo un livello alla volta.

db2gse.gse_disable_autogc

Utilizzare questa procedura memorizzata per eliminare o disabilitare temporaneamente i trigger utilizzati per la sincronizzazione di una colonna spaziale con le colonne di attributi associate. Ad esempio, è preferibile disabilitare i trigger durante il geocoding dei valori nelle colonne di attributi in modalità batch.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni, privilegi o serie di privilegi:

- Autorizzazione SYSADM o DBADM per il database che contiene la tabella in cui sono definiti i trigger da eliminare o da disabilitare temporaneamente.
- Privilegio CONTROL per la tabella.
- Privilegi ALTER e UPDATE per questa tabella.

Nota: Per i privilegi CONTROL e ALTER, è necessario disporre di autorizzazione DROPIN per lo schema DB2GSE.

Parametri:

Tabella 76. Parametri di input per la procedura memorizzata db2gse.gse_disable_autogc.

Nome	Tipo di dati	Descrizione
modo_oper	SMALLINT	<p>Indica se i trigger devono essere eliminati o disabilitati temporaneamente.</p> <p>I trigger eliminati non hanno effetto sulle istruzioni SQL.</p> <p>I trigger temporaneamente disabilitati possono essere ricreati senza dover specificare nuovamente la serie di parametri.</p> <p>Il parametro non può essere nullo.</p> <p>Per eliminare i trigger, utilizzare la macro GSE_AUTOGC_DROP. Per disabilitarli temporaneamente, utilizzare la macro GSE_AUTOGC_INVALIDATE. Per individuare i valori associati alle macro, fare riferimento al file db2gse.h. In ambiente AIX, il file è memorizzato nella directory \$DB2INSTANCE/sqlib/include/. In ambiente Windows NT, è memorizzato nella directory %DB2PATH%\include\.</p>
schemalivello	VARCHAR(30)	<p>Nome dello schema a cui appartiene la tabella o la vista specificata nel parametro tabellalivello.</p> <p>Il parametro può essere nullo.</p> <p>Se non viene indicato alcun valore per il parametro schemalivello, per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata db2gse.gse_disable_autogc.</p>
tabellalivello	VARCHAR(128)	<p>Nome della tabella che contiene la definizione dei trigger che si desidera eliminare o disabilitare temporaneamente.</p> <p>Il parametro non può essere nullo.</p>
colonnalivello	VARCHAR(128)	<p>Il nome della colonna abilitata per le operazioni spaziali gestita dai trigger che si desidera eliminare o disabilitare temporaneamente.</p> <p>Il parametro non può essere nullo.</p>

Risultati:

Tabella 77. Parametri di output per la procedura memorizzata db2gse.gse_disable_autogc.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_disable_db

Utilizzare questa procedura memorizzata per eliminare le risorse che consentono a DB2 Spatial Extender di memorizzare i dati spaziali e supportare le operazioni eseguite su questi dati.

Lo scopo della procedura è di aiutare l'utente nella risoluzione dei problemi che possono verificarsi dopo l'abilitazione del database per le operazioni spaziali e *prima* di aver aggiunto colonne di tabella spaziale o dati. Ad esempio, dopo aver abilitato un database per le operazioni spaziali, si desidera utilizzare DB2 Spatial Extender per un database diverso. Se non è stata definita alcuna colonna spaziale o se non sono ancora stati importati dati spaziali, è possibile richiamare la procedura memorizzata per eliminare tutte le risorse spaziali dal primo database.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database in cui si desidera eliminare le risorse DB2 Spatial Extender.

Risultati:

Tabella 78. Parametri di output per la procedura memorizzata db2gse.gse_disable_db.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_disable_sref

Utilizzare questa procedura memorizzata per eliminare un sistema di riferimento spaziale. Durante l'elaborazione della procedura memorizzata, le informazioni relative al sistema di riferimento spaziale vengono rimosse dalla vista del catalogo DB2GSE.SPATIAL_REF_SYS.

Prerequisiti:

Per eliminare un sistema di riferimento spaziale, è necessario annullare la registrazione dei livelli che lo utilizzano. In caso contrario, la richiesta di eliminazione del sistema di riferimento spaziale verrà rifiutata.

Autorizzazione:

Nessuna.

Processo:

Tabella 79. Parametri di input per la procedura memorizzata db2gse.gse_disable_sref.

Nome	Tipo di dati	Descrizione
IDSr	INTEGER	Identificativo numerico del sistema di riferimento spaziale da eliminare. Il parametro non può essere nullo.

Risultati:

Tabella 80. Parametri di output per la procedura memorizzata db2gse.gse_disable_sref.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_import_shape

Utilizzare questa procedura memorizzata per importare un file shape ESRI in un database abilitato per le operazioni spaziali. La procedura memorizzata può funzionare in base alle seguenti modalità:

- Se il file shape è destinato a una tabella esistente che dispone di una colonna di livello registrata, DB2 Spatial Extender inserirà nella tabella i dati del file.
- In caso contrario, DB2 Spatial Extender creerà una tabella con una colonna spaziale, registrerà la colonna come livello e inserirà nel livello e nelle altre colonne della tabella i dati del file.

Durante l'importazione di una serie di rappresentazioni shape ESRI, verranno ricevuti almeno due file. Tutti i file hanno lo stesso prefisso nel nome, ma estensione differente. Ad esempio, le estensioni dei due file che verranno sempre ricevuti, sono .shp e .shx.

Per ricevere i file di una serie di rappresentazioni shape, assegnare il nome comune dei file al parametro nome_file. Non specificare alcuna estensione. In tal modo, verranno importati tutti i file necessari, il file .shp, il file .shx ed altri eventuali file.

Ad esempio, la serie di rappresentazioni shape ESRI è memorizzata nei file Lakes.shp e Lakes.shx. Durante l'importazione di tali rappresentazioni, bisognerà assegnare al parametro nome_file solo il nome Lakes.

I file di trasferimento SDE hanno nome ma non estensione. Quindi, durante l'importazione di un file di trasferimento SDE, bisognerà assegnare il nome, ma non l'estensione, al parametro nome_file.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verranno caricati i dati shape importati.
- Privilegio CONTROL per la tabella.

Procedure memorizzate obsolete

Parametri:

Tabella 81. Parametri di input per la procedura memorizzata *db2gse.gse_import_shape*.

Nome	Tipo di dati	Descrizione
schemalivello	VARCHAR(30)	Nome dello schema a cui appartiene la tabella o la vista specificata nel parametro <i>tabellalivello</i> . Il parametro può essere nullo. Se non viene indicato alcun valore per il parametro <i>schema_livello</i> , per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata <i>db2gse.gse_import_shape</i> .
tabellalivello	VARCHAR(128)	Nome della tabella in cui verrà caricato il file shape importato. Il parametro non può essere nullo.
colonnalivello	VARCHAR(30)	Nome della colonna registrata come il livello in cui verranno caricati i dati shape. Il parametro non può essere nullo.
nome_file	VARCHAR(128)	Nome del file shape da importare. Il parametro non può essere nullo.
file_eccezioni	VARCHAR(128)	Nome e percorso del file in cui verranno memorizzati i file shape che non potranno essere importati. Si tratta di un nuovo file che verrà creato durante l'esecuzione della procedura memorizzata <i>db2gse.gse_import_shape</i> . Assegnare un nome file, ma non un'estensione, al parametro <i>file_eccezioni</i> . Il parametro non può essere nullo.
IDsr	INTEGER	Identificativo del sistema di riferimento spaziale da utilizzare per il livello in cui verranno caricati i dati shape. Il parametro può essere nullo. Se non viene specificato alcun identificativo, la trasformazione interna verrà impostata sulla massima risoluzione possibile per il file shape.
ambito_commit	INTEGER	Numero di record per punto di controllo. Il parametro può essere nullo.

Risultati:

Tabella 82. Parametri di output per la procedura memorizzata *db2gse.gse_import_shape*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_register_gc

Utilizzare questa procedura memorizzata per registrare un geocoder diverso da quello predefinito. Per determinare se un geocoder è già registrato, fare riferimento alla vista del catalogo DB2GSE.SPATIAL_GEOCODER.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database che contiene il geocoder registrato utilizzando questa procedura.

Parametri:

Tabella 83. Parametri di input per la procedura memorizzata db2gse.gse_register_gc.

Nome	Tipo di dati	Descrizione
IDgc	INTEGER	Identificativo numerico del geocoder che si desidera registrare. L'identificativo deve essere unico nel database. Il parametro non può essere nullo.
nome_gc	VARCHAR(64)	Breve descrizione del geocoder che si desidera registrare. LA descrizione deve essere una stringa di caratteri univoca all'interno del database. Il parametro non può essere nullo.
nome_fornitore	VARCHAR(64)	Nome del fornitore che ha fornito il geocoder che si desidera registrare. Il parametro non può essere nullo.
UDF_primario	VARCHAR(256)	Nome completo del geocoder che si desidera registrare. Il parametro non può essere nullo.
livello_precisione	INTEGER	Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento, affinché il geocoder elabori i dati di origine correttamente. Il livello di precisione può variare dall'1 al 100 per cento. Il parametro non può essere nullo.
specifiche_fornitore	VARCHAR(256)	Informazioni tecniche fornite dal fornitore; ad esempio, il nome e il percorso di un file utilizzato dal fornitore per impostare i parametri. Il parametro può essere nullo.
area_geo	VARCHAR(256)	L'area geografica da sottoporre a geocoding. Il parametro può essere nullo.
descrizione	VARCHAR(256)	Commenti del fornitore Il parametro può essere nullo.

Procedure memorizzate obsolete

Risultati:

Tabella 84. Parametri di output per la procedura memorizzata *db2gse.gse_register_gc*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_register_layer

Utilizzare questa procedura memorizzata per registrare una colonna spaziale come livello.n Durante l'elaborazione della procedura, le informazioni relative al livello da registrare vengono aggiunte alla vista del catalogo DB2GSE.GEOMETRY_COLUMNS.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Per il livello di tabella:
 - Autorizzazione SYSADM o DBADM per il database contiene la tabella a cui appartiene il livello
 - Privilegio CONTROL o ALTER per la tabella.
- Per il livello di vista:
 - Privilegio SELECT per le tabelle di base che contengono (1) i dati di indirizzo da sottoporre a geocoding per questo livello e (2) i dati spaziali ottenuti come risultato dopo l'operazione di geocoding.

Parametri:

Tabella 85. Parametri di input per la procedura memorizzata *db2gse.gse_register_layer*.

Nome	Tipo di dati	Descrizione
schemalivello	INTEGER(30)	Nome dello schema a cui appartiene la tabella o la vista specificata nel parametro <i>tabellalivello</i> . Il parametro può essere nullo. Se non viene indicato alcun valore per il parametro <i>schemalivello</i> , per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata <i>db2gse.gse_register_layer</i> .
tabellalivello	VARCHAR(128)	Nome della tabella o vista che contiene la colonna da registrare come livello. Il parametro non può essere nullo.

Tabella 85. Parametri di input per la procedura memorizzata db2gse.gse_register_layer. (Continua)

Nome	Tipo di dati	Descrizione
colonnalivello	VARCHAR(128)	<p>Nome della colonna da registrare come livello. Per quanto riguarda la tabella, se la colonna non esiste, DB2 Spatial Extender aggiungerà la colonna utilizzando l'istruzione ALTER. Per quanto riguarda la vista, è necessario che la colonna esista già.</p> <p>E' possibile specificare solo una colonna per il parametro colonnalivello. Quindi, registrando più colonne di una tabella o vista come livelli, è necessario eseguire la procedura memorizzata separatamente per ciascuna colonna.</p> <p>Il parametro non può essere nullo.</p>
nome_tipolivello	VARCHAR(64)	<p>Tipo di dati della colonna da registrare come livello. Vengono accettati solo i tipi di dati forniti da DB2 Spatial Extender. E' necessario specificare il tipo di dati in caratteri maiuscoli; ad esempio: ST_POINT</p> <p>Non è necessario specificare uno schema, in quanto viene aggiunto automaticamente.</p> <p>Il parametro non può essere nullo se la colonna appartiene a una tabella che dovrà essere creata durante l'elaborazione della procedura memorizzata. In caso contrario, se la colonna esiste già in una tabella o vista, sarà possibile annullare il parametro.</p>
IDsr	INTEGER	<p>Identificativo del sistema di riferimento spaziale utilizzato per questo livello.</p> <p>Il parametro non può essere se riferito a un livello di tabella. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di un livello di vista.</p>
schema_geo	VARCHAR(30)	<p>Lo schema della tabella che indica la vista a cui appartiene la colonna. Questo parametro viene applicato se la colonna di vista viene registrata come livello.</p> <p>Se la colonna di vista viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di tabella come livello.</p> <p>Le viste basate su più di una tabella di base o viste di tipo diverso non sono supportate da questo parametro.</p> <p>Se non viene indicato alcun valore per il parametro schema_geo, per impostazione predefinita, verrà utilizzato il valore del parametro schemalivello.</p>

Procedure memorizzate obsolete

Tabella 85. Parametri di input per la procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
tabella_geo	VARCHAR(128)	<p>Il nome della tabella che indica la vista a cui appartiene la colonna. Questo parametro viene applicato se la colonna di vista viene registrata come livello.</p> <p>Le viste basate su più di una tabella di base o viste di tipo diverso non sono supportate da questo parametro.</p> <p>Se la colonna di vista viene registrata come livello, il parametro non può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di tabella come livello.</p>
colonna_geo	VARCHAR(128)	<p>Il nome della tabella che indica questa colonna di vista. Questo parametro viene applicato se la colonna di vista viene registrata come livello.</p> <p>Le viste basate su più di una tabella di base o viste di tipo diverso non sono supportate da questo parametro.</p> <p>Se la colonna di vista viene registrata come livello, il parametro non può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di tabella come livello.</p>
n_attributi	SMALLINT	<p>Numero delle colonne che contengono i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p>
nome_attr_1	VARCHAR(128)	<p>Nome della prima colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Se si desidera utilizzare il geocoder predefinito, è necessario memorizzare gli indirizzi nella colonna <i>nome_attr_1</i>.</p>
nome_attr_2	VARCHAR(128)	<p>Nome della seconda colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Se si desidera utilizzare il geocoder predefinito, è necessario memorizzare i nomi delle città nella colonna <i>nome_attr_2</i>.</p>

Tabella 85. Parametri di input per la procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
nome_attr_3	VARCHAR(128)	<p>Nome della terza colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Se si desidera utilizzare il geocoder predefinito, è necessario memorizzare i nomi delle abbreviazioni degli stati nella colonna nome_attr_3.</p>
nome_attr_4	VARCHAR(128)	<p>Nome della quarta colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Se si desidera utilizzare il geocoder predefinito, è necessario memorizzare i codici postali nella colonna nome_attr_4.</p>
nome_attr_5	VARCHAR(128)	<p>Nome della quinta colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Il geocoder predefinito ignora la colonna nome_attr_5.</p>
nome_attr_6	VARCHAR(128)	<p>Nome della sesta colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Il geocoder predefinito ignora la colonna nome_attr_6.</p>
nome_attr_7	VARCHAR(128)	<p>Nome della settima colonna che contiene i dati di origine da sottoporre a geocoding per questo livello.</p> <p>Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello.</p> <p>Il geocoder predefinito ignora la colonna nome_attr_7.</p>

Procedure memorizzate obsolete

Tabella 85. Parametri di input per la procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
nome_attr_8	VARCHAR(128)	Nome dell'ottava colonna che contiene i dati di origine da sottoporre a geocoding per questo livello. Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello. Il geocoder predefinito ignora la colonna <i>nome_attr_8</i> .
nome_attr_9	VARCHAR(128)	Nome della nona colonna che contiene i dati di origine da sottoporre a geocoding per questo livello. Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello. Il geocoder predefinito ignora la colonna <i>nome_attr_9</i> .
nome_attr_10	VARCHAR(128)	Nome della decima colonna che contiene i dati di origine da sottoporre a geocoding per questo livello. Se la colonna di tabella viene registrata come livello, il parametro può essere annullato. DB2 Spatial Extender ignora il parametro se si effettua la registrazione di una colonna di vista come livello. Il geocoder predefinito ignora la colonna <i>nome_attr_10</i> .

Risultati:

Tabella 86. Parametri di output per la procedura memorizzata *db2gse.gse_register_layer*.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

Limitazioni:

La procedura memorizzata non funziona con i seguenti tipi di tabella:

- A = Alias
- H = Tabella gerarchica
- N = Soprannome
- S = Tabella di riepilogo
- U = Tabella di immissione
- W = Vista di immissione

Inoltre, sono valide le seguenti limitazioni:

- Se si desidera registrare la colonna di vista come livello, è necessario che la colonna si basi su una colonna di tabella già registrata come livello.
- Non è possibile inserire i dati da sottoporre a geocoding per il livello da registrare, in più di dieci colonne di attributi.

db2gse.gse_run_gc

Utilizzare questa procedura memorizzata per eseguire un geocoder in modalità batch.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Autorizzazione SYSADM o DBADM per il database contiene la tabella in cui verrà eseguito il geocoder specificato.
- Privilegio CONTROL o UPDATE per questa tabella.

Parametri:

Tabella 87. Parametri di input per la procedura memorizzata db2gse.gse_run_gc.

Nome	Tipo di dati	Descrizione
schemalivello	VARCHAR(30)	Nome dello schema a cui appartiene la tabella o la vista specificata nel parametro tabellalivello. Il parametro può essere nullo. Se non viene indicato alcun valore per il parametro schemalivello, per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata db2gse.gse_run_gc.
tabellalivello	VARCHAR(128)	Nome della tabella che contiene la colonna in cui si desidera inserire i dati sottoposti a geocoding. Il parametro non può essere nullo.
colonnalivello	VARCHAR(128)	Nome della colonna in cui si desidera inserire i dati sottoposti a geocoding. Il parametro non può essere nullo.
IDgc	INTEGER	Identificativo numerico del geocoder che si desidera eseguire. Il parametro può essere nullo. Per individuare gli identificatori dei geocoder registrati, fare riferimento alla vista del catalogo DB2GSE.SPATIAL_GEOCODER.
livello_precisione	INTEGER	Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento, affinché il geocoder elabori i dati di origine correttamente. Il parametro può essere nullo. Il livello di precisione può variare dall'1 al 100 per cento.

Procedure memorizzate obsolete

Tabella 87. Parametri di input per la procedura memorizzata db2gse.gse_run_gc. (Continua)

Nome	Tipo di dati	Descrizione
specifiche_fornitore	VARCHAR(256)	Informazioni tecniche fornite dal fornitore; ad esempio, il nome e il percorso di un file utilizzato dal fornitore per impostare i parametri. Il parametro può essere nullo.
clausola_where	VARCHAR(256)	Il testo della clausola WHERE. Definisce una limitazione per la serie di record da sottoporre a geocoding. La clausola può fare riferimento ad una qualsiasi colonna di attributo della tabella in cui viene eseguito il geocoder. Il parametro può essere nullo.
ambito_commit	INTEGER	Numero di record per punto di controllo. Il parametro può essere nullo.

Risultati:

Tabella 88. Parametri di output per la procedura memorizzata db2gse.gse_run_gc.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_unregist_gc

Utilizzare questa procedura memorizzata per annullare la registrazione di un geocoder diverso da quello predefinito. Le informazioni sul geocoder di cui si desidera annullare la registrazione, sono contenute nella vista del catalogo DB2GSE.SPATIAL_GEOCODER.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre dell'autorizzazione SYSADM o DBADM per il database che contiene il geocoder di cui si desidera annullare la registrazione.

Parametri:

Tabella 89. Parametri di input per la procedura memorizzata db2gse.gse_unregist_gc.

Nome	Tipo di dati	Descrizione
IDgc	INTEGER	L'identificativo del geocoder di cui si desidera annullare la registrazione. Il parametro non può essere nullo.

Risultati:

Tabella 90. Parametri di output per la procedura memorizzata db2gse.gse_unregist_gc.

Nome	Tipo di dati	Descrizione
codice_msg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testo_msg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

db2gse.gse_unregist_layer

Utilizzare questa procedura memorizzata per annullare la registrazione di livello. La procedura annulla la registrazione:

- Eliminando la definizione del livello dalle tabelle del catalogo di DB2 Spatial Extender.
- Eliminando il vincolo di controllo inserito da DB2 Spatial Extender nella tabella di base del livello per assicurarsi che i dati spaziali del livello siano conformi ai requisiti del sistema di riferimento spaziale utilizzato dal livello.
- Eliminando i trigger utilizzato per aggiornare la colonna spaziale in caso di aggiunta, modifica o rimozioni dei dati di indirizzo.

Quando i dati di una riga della tabella vengono sottoposti a geocoding, i dati spaziali ottenuti come risultato vengono inseriti nella stessa riga. Se la riga viene eliminata, saranno eliminati anche i dati spaziali e di indirizzo. I trigger non eliminano i dati spaziali. Durante l'elaborazione della procedura memorizzata, le informazioni relative al livello vengono rimosse dalla vista del catalogo DB2GSE.GEOMETRY_COLUMNS.

Autorizzazione:

L'ID utente utilizzato per richiamare la procedura memorizzata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Per il livello di tabella:
 - Autorizzazione SYSADM o DBADM per il database contiene la tabella di base del livello
 - Privilegio CONTROL o ALTER per la tabella.
- Per il livello di vista:
 - Privilegio SELECT per le tabelle di base che contengono (1) i dati di indirizzo da sottoporre a geocoding per questo livello e (2) i dati spaziali ottenuti come risultato dopo l'operazione di geocoding.

Procedure memorizzate obsolete

Parametri:

Tabella 91. Parametri di input per la procedura memorizzata db2gse.gse_unregist_layer.

Nome	Tipo di dati	Descrizione
schemalivello	VARCHAR(30)	Nome dello schema a cui appartiene la tabella specificata nel parametro tabellalivello. Il parametro può essere nullo. Se non viene indicato alcun valore per il parametro schemalivello, per impostazione predefinita verrà utilizzato l'ID utente utilizzato per richiamare la procedura memorizzata db2gse.gse_unregister_layer. E' necessario specificare il nome dello schema, della tabella, della vista, della colonna o del livello che si desidera assegnare ad un parametro, in caratteri maiuscoli.
tabellalivello	VARCHAR(128)	Nome della tabella che contiene la colonna specificata dal parametro colonnalivello. Il parametro non può essere nullo.
colonnalivello	VARCHAR(128)	Nome della colonna spaziale definita come il livello di cui si desidera annullare la registrazione. Il parametro non può essere nullo. E' possibile specificare solo un livello per il parametro colonnalivello. Quindi, registrando più livelli in una tabella o vista, è necessario eseguire la procedura memorizzata separatamente per ciascun livello.

Risultati:

Tabella 92. Parametri di output per la procedura memorizzata db2gse.gse_unregist_layer.

Nome	Tipo di dati	Descrizione
codicemsg	INTEGER	Codice associato ai messaggi restituiti dal chiamante della procedura memorizzata.
testomsg	VARCHAR(1024)	Messaggio di errore completo creato dal server DB2 Spatial Extender.

Limitazione:

Se una colonna definita come vista si basa su una colonna di tabella definita come livello di tabella, non sarà possibile annullare la registrazione della tabella fino a quando non viene annullata la registrazione del livello di vista.

Appendice B. Viste del catalogo obsolete

In questa sezione vengono descritte le viste del catalogo obsolete.

Nota: Suggerimento: Sviluppare tutte le nuove applicazioni con le viste definite in DB2 Spatial Extender Versione 8. E' preferibile aggiornare anche tutte le applicazioni in modo utilizzino le viste definite nella Versione 8. Dopo aver eseguito la migrazione alla Versione 8, non sarà più possibile utilizzare le applicazioni che fanno riferimento alle tabelle del catalogo definite nella Versione 7.

DB2GSE.COORD_REF_SYS

Quando si abilita un database per le operazioni spaziali, DB2 Spatial Extender registra i sistemi di coordinate che è possibile utilizzare in una tabella di catalogo. Le colonne selezionate dalla tabella comprendono la vista del catalogo DB2GSE.COORD_REF_SYS, descritta nella seguente tabella.

Tabella 93. Colonne nella vista del catalogo DB2GSE.COORD_REF_SYS

Nome	Tipo di dati	Annulabile?	Contenuto
CSID	INTEGER	Sì	Identificativo numerico per questo sistema di coordinate. Se il sistema di coordinate è stato creato utilizzando l'interfaccia di gestione V8, non verrà registrato alcun CSID e verrà utilizzato un valore nullo
CS_NAME	VARCHAR(64)	No	Nome del sistema di coordinate.
AUTH_NAME	VARCHAR(256)	Sì	Il nome dell'organizzazione che ha compilato il sistema di coordinate; ad esempio l'European Petroleum Survey Group (EPSG).
AUTH_SRID	INTEGER	Sì	Identificativo numerico assegnato al sistema di coordinate dall'organizzazione indicata nella colonna AUTH_NAME.
DESC	VARCHAR(256)	Sì	Descrizione del sistema di coordinate.
SRTEXT	VARCHAR(2048)	No	Testo per il sistema di coordinate.

DB2GSE.GEOMETRY_COLUMNS

Una volta creato un livello, DB2 Spatial Extender lo registra memorizzandone l'identificativo e le informazioni ad esso relative in una tabella del catalogo. Le colonne selezionate da questa tabella comprendono la vista del catalogo DB2GSE.GEOMETRY_COLUMNS, descritta nella seguente tabella.

Tabella 94. Colonne nella vista del catalogo DB2GSE.GEOMETRY_COLUMNS

Nome	Tipo di dati	Annulabile?	Contenuto
LAYER_CATALOG	VARCHAR(30)	Sì	NULLO. DB2 Spatial Extender non contiene LAYER_CATALOG.
LAYER_SCHEMA	VARCHAR(30)	No	Schema della tabella o vista che contiene la colonna registrata come livello.

Viste del catalogo obsolete

Tabella 94. Colonne nella vista del catalogo DB2GSE.GEOMETRY_COLUMNS (Continua)

Nome	Tipo di dati	Annullabile?	Contenuto
LAYER_TABLE	VARCHAR(128)	No	Nome della tabella o vista che contiene la colonna registrata come livello.
LAYER_COLUMN	VARCHAR(128)	No	Nome della colonna registrata come livello.
GEOMETRY_TYPE	INTEGER	Sì	Tipo di dati della colonna registrata come livello. Se la colonna dispone di un tipo secondario definito dall'utente di uno dei tipi di forma geometrica definito da Spatial Extender, questo valore sarà nullo.
SRID	INTEGER	No	Identificativo del sistema di riferimento spaziale utilizzato per i valori nella colonna registrata in questo livello.
STORAGE_TYPE	INTEGER	Sì	NULLO.

DB2GSE.SPATIAL_GEOCODER

I geocoder disponibili vengono registrati in una tabella del catalogo. Le colonne selezionate da questa tabella comprendono la vista del catalogo DB2GSE.SPATIAL_GEOCODER, descritta nella seguente tabella.

Tabella 95. Colonne nella vista del catalogo DB2GSE.SPATIAL_GEOCODER

Nome	Tipo di dati	Annullabile?	Contenuto
GCID	INTEGER	No	Identificativo numerico del geocoder.
GC_NAME	VARCHAR(64)	No	Identificativo del nome del geocoder.
VENDOR_NAME	VARCHAR(128)	No	Nome del fornitore che ha fornito il geocoder.
PRIMARY_UDF	VARCHAR(256)	No	Nome completo del geocoder.
PRECISION_LEVEL	INTEGER	No	Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento, affinché il geocoder li elabori correttamente.
VENDOR_SPECIFIC	VARCHAR(256)	Sì	Percorso e nome di un file che il fornitore può utilizzare per impostare parametri speciali supportati dal geocoder.
GEO_AREA	VARCHAR(256)	Sì	Area geografica che contiene le posizioni da sottoporre a geocoding.
DESCRIPTION	VARCHAR(256)	Sì	Descrizione del geocoder.

DB2GSE.SPATIAL_REF_SYS

Una volta creato un sistema di riferimento spaziale, DB2 Spatial Extender lo registra memorizzandone l'identificativo e le informazioni ad esso relative in una tabella del catalogo. Le colonne selezionate dalla tabella comprendono la vista del catalogo DB2GSE.SPATIAL_REF_SYS, descritta nella seguente tabella.

Tabella 96. Colonne nella vista del catalogo DB2GSE.SPATIAL_REF_SYS

Nome	Tipo di dati	Annullabile?	Contenuto
SRID	INTEGER	No	Identificativo definito dall'utente per questo sistema di riferimento spaziale.
SR_NAME	VARCHAR(64)	No	Nome del sistema di riferimento spaziale.

Tabella 96. Colonne nella vista del catalogo DB2GSE.SPATIAL_REF_SYS (Continua)

Nome	Tipo di dati	Annullabile?	Contenuto
CSID	INTEGER	No	Identificativo numerico del sistema di coordinate su cui si basa il sistema di riferimento spaziale.
CS_NAME	VARCHAR(64)	No	Nome del sistema di coordinate su cui si basa il sistema di riferimento spaziale.
AUTH_NAME	VARCHAR(256)	Sì	Nome dell'organizzazione che stabilisce gli standard per il sistema di riferimento spaziale.
AUTH_SRID	INTEGER	Sì	Identificativo numerico che l'organizzazione indicata nella colonna AUTH_NAME assegna al sistema di riferimento spaziale.
SRTEXT	VARCHAR(2048)	No	Testo per il sistema di riferimento spaziale.
FALSEX	FLOAT	No	Un numero che, sottratto da un valore di coordinata X negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero).
FALSEY	FLOAT	No	Un numero che, sottratto da un valore di coordinata Y negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero).
XYUNITS	FLOAT	No	Un numero che, moltiplicato per una coordinata X decimale o per una coordinata Y decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit.
FALSEZ	FLOAT	No	Un numero che, sottratto da un valore di coordinata Z negativo, dia come risultato un numero non negativo (ovvero un numero positivo o zero).
ZUNITS	FLOAT	No	Un numero che, moltiplicato per una coordinata Z decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit.
FALSEM	FLOAT	No	Un numero che, sottratto da una misura negativa, dia come risultato un numero non negativo (ovvero un numero positivo o zero).
MUNITS	FLOAT	No	Un numero che, moltiplicato per una misura decimale, restituisca un numero intero che possa essere memorizzato come voce di dati a 32-bit.

Viste del catalogo obsolete

Appendice C. Funzioni spaziali obsolete

In questa sezione vengono descritte le funzioni obsolete. La tabella seguente contiene tutte le funzioni spaziali obsolete e le funzioni con le quali sono state sostituite nella Versione 8.

Tabella 97. Funzioni obsolete e nuove funzioni corrispondenti.

Funzione obsoleta	Nuova funzione
AsShape	ST_AsShape
GeometryFromShape	ST_Geometry
Is3D	ST_Is3D
IsMeasured	ST_IsMeasured
LineFromShape	ST_LineString
LocateAlong	ST_FindMeasure
LocateBetween	ST_MeasureBetween
M	ST_M
MLine FromShape	ST_MultiLineString
MPointFromShape	ST_MultiPoint
MPolyFromShape	ST_MultiPolygon
PointFromShape	ST_Point
PolyFromShape	ST_Polygon
ShapeToSQL	ST_Geometry
ST_GeomFromText	ST_Geometry
ST_GeomFromWKB	ST_Geometry
ST_LineFromText	ST_LineString
ST_LineFromWKB	ST_LineString
ST_MLineFromText	ST_MultiLineString
ST_MLineFromWKB	ST_MultiLineString
ST_MPointFromText	ST_MultiPoint
ST_MPointFromWKB	ST_MultiPoint
ST_MPolyFromText	ST_MultiPolygon
ST_MPolyFromWKB	ST_MultiPolygon
ST_OrderingEquals	
ST_Point(Double, Double, db2gse.coordref)	ST_Point(Double, Double, Integer)
ST_PointFromText	ST_Point
ST_PolyFromText	ST_Polygon
ST_PolyFromWKB	ST_Polygon
ST_Transform(Double, Double, db2gse.coordref)	ST_Transform(ST_Geometry, Integer)
ST_SymmetricDiff	ST_SymDifference
Z	ST_Z

AsShape

Scopo:

AsShape restituisce un BLOB da una oggetto forma geometrica.

Formato:

```
db2gse.AsShape(g db2gse.ST_Geometry)
```

Risultati:

BLOB(1m)

GeometryFromShape

Scopo:

GeometryFromShape parte da una forma e un identificativo di sistema di riferimento spaziale e restituisce un oggetto forma geometrica.

Formato:

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), SRID db2gse.coordref)
```

Risultati:

db2gse.ST_Geometry

Is3d

Scopo:

Is3d parte da un oggetto di forma geometrica e restituisce 1 se l'oggetto ha coordinate 3D; in caso contrario restituisce 0.

Formato:

```
db2gse.Is3d(g db2gse.ST_Geometry)
```

Risultati:

Integer

IsMeasured

Scopo:

IsMeasured parte da un oggetto di forma geometrica e restituisce 1 se l'oggetto ha misure; in caso contrario restituisce 0.

Formato:

```
db2gse.IsMeasured(g db2gse.ST_Geometry)
```

Risultati:

Integer

LineFromShape

Scopo:

LineFromShape parte dalla forma di un punto e da un identificativo di sistema di riferimento spaziale e restituisce una linea.

Formato:

```
db2gse.Line FromShape(ShapeLineString Blob(1M), SRID db2gse.coordref)
```

Risultati:

db2gse.ST_LineString

LocateAlong

Scopo:

LocateAlong parte da un oggetto forma geometrica e da una misura e restituisce la serie di punti rilevata nella misura come multipunto.

Se LocateAlong utilizza un multipunto e una misura come input, e se il multipunto non comprende la misura, LocateAlong restituisce POINT EMPTY.

Formato:

```
db2gse.LocateAlong(g db2gse.ST_Geometry, measure Double)
```

Risultati:

db2gse.ST_Geometry

LocateBetween

Scopo:

LocateBetween parte da un oggetto forma geometrica e due posizioni con misure e restituisce la forma geometrica che rappresenta la serie di percorsi non collegati tra loro, tra a le due posizioni.

Formato:

```
db2gse.LocateBetween(g db2gse.ST_Geometry, measure Double, measure Double)
```

Risultati:

db2gse.ST_Geometry

M

Scopo:

M parte da un punto e ne restituisce la misura.

Formato:

db2gse.M(p db2gse.ST_Point)

Risultati:

Double

MLine FromShape

Scopo:

MLine FromShape parte da una forma di tipo multilinea e da un identificativo di sistema di riferimento spaziale e restituisce una multilinea.

Formato:

db2gse.MLineFromShape(ShapeMultiLineString Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiLineString

MPointFromShape

Scopo:

MPointFromShape parte da una forma di tipo multipunto e da un identificativo di sistema spaziale e restituisce un multipunto.

Formato:

db2gse.MPointFromShape(ShapeMultiPoint BLOB(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiPoint

MPolyFromShape

Scopo:

MPolyFromShape parte da una forma di tipo multipoligono e da un identificativo di sistema spaziale e restituisce un multipoligono.

Formato:

db2gse.MPolyFromShape(ShapeMultiPolygon Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiPolygon

PointFromShape

Scopo:

PointFromShape parte da una forma di tipo punto e da un identificativo di sistema spaziale e restituisce un punto.

Formato:

db2gse.PointFromShape(db2gse.ShapePoint blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_Point

PolyFromShape

Scopo:

PolyFromShape parte da una forma di tipo poligono e da un identificativo di sistema spaziale e restituisce un poligono.

Formato:

db2gse.PolyFromShape (ShapePolygon Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_Polygon

ShapeToSQL

Scopo:

ShapeToSQL crea la rappresentazione shape di un valore db2gse.ST_Geometry. Il valore SRID 0 viene utilizzato automaticamente.

Formato:

db2gse.ShapeToSQL(ShapeGeometry blob(1M))

Risultati:

db2gse.ST_Geometry

ST_GeomFromText

Scopo:

ST_GeomFromText parte da una rappresentazione WKT e da un sistema di riferimento spaziale e restituisce un oggetto forma geometrica.

Formato:

```
db2gse.ST_GeomFromText(geometryTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Risultati:

```
db2gse.ST_Geometry
```

ST_GeomFromWKB

Scopo:

ST_GeomFromWKB parte da una rappresentazione WKB e da un sistema di riferimento spaziale e restituisce un oggetto forma geometrica.

Formato:

```
db2gse.ST_GeomFromWKB(WKBGeometry Blob(1M), SRID db2gse.coordref)
```

Risultati:

```
db2gse.ST_Geometry
```

ST_LineFromText

Scopo:

ST_LineFromText parte da una rappresentazione WKT di tipo linea e da un identificativo di sistema di riferimento spaziale e restituisce una linea.

Formato:

```
db2gse.ST_LineFromText(lineStringTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Risultati:

```
db2gse.ST_LineString
```

ST_LineFromWKB

Scopo:

ST_LineFromWKB parte da una rappresentazione WKB di tipo linea e da un identificativo di sistema di riferimento spaziale e restituisce una linea.

Formato:

db2gse.ST_LineFromWKB(WKBLineString Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_LineString

ST_MLineFromText

Scopo:

ST_MLineFromText parte da una rappresentazione WKT di tipo multilinea e da un identificativo di sistema di riferimento spaziale e restituisce una multilinea.

Formato:

db2gse.ST_MLineFromText(multiLineStringTaggedText String, SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiLineString

ST_MLineFromWKB

Scopo:

ST_MLineFromWKB parte da una rappresentazione WKB di tipo multilinea e da un identificativo di sistema di riferimento spaziale e restituisce una multilinea.

Formato:

db2gse.ST_MLineFromWKB(WKBMultiLineString Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiLineString

ST_MPointFromText

Scopo:

ST_MPointFromText parte da una rappresentazione WKT di tipo multipunto e da un identificativo di sistema di riferimento spaziale e restituisce un multipunto.

Formato:

db2gse.ST_MPointFromText(multiPointTaggedText Varchar(4000), SRID db2gse.coordref)

Funzioni spaziali obsolete

Risultati:

db2gse.ST_MultiPoint

ST_MPointFromWKB

Scopo:

ST_MPointFromWKB parte da una rappresentazione WKB di tipo multipunto e da un identificativo di sistema di riferimento spaziale e restituisce un multipunto.

Formato:

db2gse.ST_MPointFromWKB(WKBMultiPoint Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiPoint

ST_MPolyFromText

Scopo:

ST_MPolyFromText parte da una rappresentazione WKT di tipo multipoligono e da un identificativo di sistema di riferimento spaziale e restituisce un multipoligono.

Questa funzione non può utilizzare come input un multipoligono che contiene più poligoni aventi le stesse coordinate.

Formato:

db2gse.ST_MPolyFromText(multiPolygonTaggedText Varchar(4000), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiPolygon

ST_MPolyFromWKB

Scopo:

ST_MPolyFromWKB parte da una rappresentazione WKB di tipo multipoligono e da un identificativo di sistema di riferimento spaziale e restituisce un multipoligono.

Formato:

db2gse.ST_MPolyFromWKB(WKBMultiPolygon Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_MultiPolygon

ST_OrderingEquals

Scopo:

ST_OrderingEquals paragona due forme geometriche e restituisce 1 (TRUE) se le forme sono uguali e le coordinate hanno lo stesso ordine; in caso contrario restituisce 0 (FALSE).

Formato:

db2gse.ST_OrderingEquals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)

Risultati:

Integer

ST_Point

Scopo:

ST_Point restituisce un ST_Point, partendo da una coordinata-x, una coordinata-y e un riferimento spaziale.

Formato:

db2gse.ST_Point(X Double, Y Double, SRID db2gse.coordref)

Risultati:

db2gse.ST_Point

ST_PointFromText

Scopo:

ST_PointFromText parte da una rappresentazione WKT di tipo punto e da un identificativo di sistema di riferimento spaziale e restituisce un punto.

Formato:

db2gse.ST_PointFromText(pointTaggedText Varchar(4000), SRID db2gse.coordref)

Risultati:

db2gse.ST_Point

ST_PolyFromText

Scopo:

Funzioni spaziali obsolete

ST_PolyFromText parte da una rappresentazione WKT di tipo multipoligono e da un identificativo di sistema di riferimento spaziale e restituisce un multipoligono.

Formato:

db2gse.ST_PolyFromText(polygonTaggedText Varchar(4000), SRID db2gse.coordref)

Risultati:

db2gse.ST_Polygon

ST_PolyFromWKB

Scopo:

ST_PolyFromWKB parte da una rappresentazione WKB di tipo poligono e da un identificativo di sistema di riferimento spaziale e restituisce un poligono.

Formato:

db2gse.ST_PolyFromWKB(WKBPolygon Blob(1M), SRID db2gse.coordref)

Risultati:

db2gse.ST_Polygon

ST_Transform

Scopo:

ST_Transform associa una forma geometrica a un sistema di riferimento spaziale diverso dal sistema spaziale a cui la forma geometrica è associata.

Formato:

db2gse.ST_Transform(g db2gse.ST_Geometry, SRID db2gse.coordref)

Risultati:

db2gse.ST_Geometry

ST_SymmetricDiff

Scopo:

ST_SymmetricDiff parte da due forme geometriche e restituisce un oggetto forma geometrica che rappresenta la differenza simmetrica degli oggetti di origine.

La funzione ST_SymmetricDiff restituisce la differenza simmetrica (la logica Booleana XOR dello spazio) di due forme geometriche intersecanti aventi la stessa dimensione. Se le forme geometriche sono uguali, ST_SymmetricDiff restituisce una forma geometrica vuota. In caso contrario, parte di una o di entrambe, risulterà esterna all'area dell'intersezione. ST_SymmetricDiff restituisce le parti non intersecanti come insieme; ad esempio come multipoligono.

Se ST_SymmetricDiff utilizza come input forme geometriche di dimensioni diverse, restituirà un valore nullo.

Formato:

db2gse.ST_SymmetricDiff(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)

Risultati:

db2gse.ST_Geometry

Z

Scopo:

Z parte da un punto e restituisce la coordinata Z corrispondente.

Formato:

db2gse.Z(p db2gse.ST_Point)

Risultati:

Double

Riferimenti correlati:

- "ST_AsShape" a pagina 350
- "ST_MeasureBetween, ST_LocateBetween" a pagina 432
- "ST_EnvIntersects" a pagina 379
- "ST_FindMeasure o ST_LocateAlong" a pagina 385
- "ST_Geometry" a pagina 393
- "ST_Is3d" a pagina 407
- "ST_LineString" a pagina 419
- "ST_M" a pagina 422
- "ST_MultiLineString" a pagina 450
- "ST_MultiPoint" a pagina 451
- "ST_MultiPolygon" a pagina 453
- "ST_Point" a pagina 465
- "ST_Polygon" a pagina 475
- "ST_SymDifference" a pagina 484
- "ST_Transform" a pagina 495
- "ST_Z" a pagina 506

Funzioni spaziali obsolete

Informazioni particolari

E' possibile che negli altri paesi l'IBM non offra i prodotti, le funzioni o i servizi illustrati in questo documento. Consultare il rappresentante IBM locale per informazioni sui prodotti o sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possono essere utilizzati. In sostituzione a quelli forniti dall'IBM, è possibile usare prodotti, programmi o servizi funzionalmente equivalenti che non comportino violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. E' comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri prodotti, programmi o servizi non IBM.

L'IBM può avere brevetti o domande di brevetti in corso relativi a quanto trattato nella presente pubblicazione. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. Chi desiderasse ricevere informazioni relative alle licenze può rivolgersi per iscritto a:

IBM Director of Commercial Relations
IBM Corporation
Schoenaicher Str. 220
D-7030 Boeblingen
Deutschland

Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:

L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "AS IS" SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZATA ED IDONEITA' AD UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni, quindi, la presente dichiarazione potrebbe non essere a voi applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche verranno incorporate nelle nuove edizioni della pubblicazione. L'IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto nel manuale in qualsiasi momento e senza preavviso.

Tutti i riferimenti a siti Web non dell'IBM sono forniti unicamente a scopo di consultazione. I materiali contenuti in tali siti Web non fanno parte di questo prodotto e l'utente si assume ogni rischio relativo al loro utilizzo.

L'IBM può utilizzare o divulgare le informazioni ricevute dagli utenti secondo le modalità ritenute appropriate, senza alcun obbligo nei loro confronti.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni allo scopo di consentire: (i) uno scambio di informazioni tra programmi indipendenti e altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue

Markham, Ontario
L6G 1C7
CANADA

Queste informazioni possono essere rese disponibili, secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento in addebito.

Il programma su licenza descritto in questo manuale e tutto il materiale su licenza ad esso relativo sono forniti dall'IBM nel rispetto delle condizioni previste dalla licenza d'uso.

Qualsiasi informazione relativa alle prestazioni è stata verificata in un ambiente controllato. Di conseguenza l'utilizzo del prodotto in ambienti operativi diversi può comportare risultati sensibilmente diversi. Alcune rilevazioni possono essere state effettuate su sistemi a livello di sviluppo e non si garantisce in alcun modo, dunque, che siano uguali alle rilevazioni eseguite sui vari sistemi disponibili. Inoltre, è possibile che ad alcune di queste rilevazioni si sia pervenuti tramite estrapolazione. I risultati reali potrebbero variare. E' necessario che gli utenti confrontino i dati in base agli ambienti utilizzati.

Le informazioni relative a prodotti non IBM sono state ottenute dai fornitori di tali prodotti. L'IBM non ha verificato tali prodotti e non può garantire l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Le dichiarazioni relative a futuri intenti o obiettivi IBM sono soggette a modifiche senza preavviso.

Questa pubblicazione contiene esempi di dati e prospetti usati quotidianamente nelle operazioni aziendali. Pertanto può contenere nomi di persone, società, marchi e prodotti. Tutti i nomi contenuti nel manuale sono fittizi e ogni riferimento a nomi ed indirizzi reali è puramente casuale.

LICENZA RELATIVA AI DIRITTI D'AUTORE:

Queste informazioni contengono programmi applicativi di esempio in lingua originale che illustrano le tecniche di programmazione su diverse piattaforme operative. Potete copiare, modificare e distribuire questi esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in modo conforme alle API (Application Programming Interface) a seconda della piattaforma operativa per cui tali esempi di programmi sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o assicurare l'affidabilità, la praticità o il funzionamento di questi programmi.

Ogni copia o ogni parte di questi programmi campione o di qualsiasi lavoro derivato, deve includere la seguente informativa relativa ai diritti d'autore:

© (*nome della vostra società*) (*anno*). Parti di questo codice derivano dalla IBM Corp. Sample Programs. Programmi di esempio © Copyright IBM Corp. *_inserire l'anno o gli anni_*. Tutti i diritti riservati.

Marchi

I seguenti termini sono marchi della International Business Machines Corporation degli Stati Uniti o di altri paesi e sono stati utilizzati in minimo un documento della libreria DB2 UDB.

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extenders	SQL/400
DB2 OLAP Server	SQL/DS
DB2 Information Integrator	System/370
DB2 Query Patroller	System/390
Server DB2 Universal	SystemView
Distributed Relational Database Architecture	Tivoli
DRDA	VisualAge
eServer	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
IBM	WebSphere
IMS	WIN-OS/2
IMS/ESA	z/OS
	zSeries

I seguenti termini sono marchi di altre società e sono stati utilizzati in minimo un documento della libreria DB2 UDB:

Microsoft, Windows, Windows NT e il logo Windows sono marchi della Microsoft Corporation negli Stati Uniti e/o altri paesi.

Intel e Pentium sono marchi della Intel Corporation negli Stati Uniti e/o altri paesi.

Java e tutti i marchi a base Java sono marchi della Sun Microsystems, Inc. negli Stati Uniti e/o altri paesi.

UNIX è un marchio registrato della The Open Group negli Stati Uniti e/o altri paesi.

Nomi di altri prodotti, società e servizi possono essere marchi di altre società.

Indice analitico

Numerico

- 180° meridiano
 - linee che lo attraversano 197
 - MBC che lo attraversano 206
- 180° meridiano, linee che lo attraversano 197

A

- abilitazione
 - operazioni spaziali 53, 54
- abilitazione licenza geodetica 165
- AIX
 - installazione
 - DB2 Spatial Extender 29
- analisi degli indici
 - utilizzo di Index Advisor 112
- applicazioni
 - applicazioni spaziali
 - inclusione file di intestazione 133
 - programma di esempio 135
 - Spatial Extender
 - richiamo procedure memorizzate 134
 - spaziali 133
- applicazioni spaziali
 - inclusione file di intestazione 133
 - procedure memorizzate
 - richiamo dalle applicazioni 134
- ArcExplorer
 - utilizzo di un'interfaccia 121
- AsShape, funzione spaziale obsoleta 561
- associazione dati
 - Spatial Extender 42
- attività
 - impostazione di Spatial Extender 17
- attributi ST_Geometry
 - differenze geodetiche 197

C

- Centro di controllo
 - messaggi 151
- cintura equatoriale
 - poligoni che la rappresentano 197
- circonferenze
 - definizione di regioni geodetiche 161
 - descrizione 13
- CLP (command line processor)
 - comandi di Spatial Extender 125
 - messaggi 149
- colonne
 - dati spaziali nelle 83
- colonne spaziali
 - creazione 83
 - geocoding 93
 - registrazione nel sistema di riferimento spaziale 85
 - riempimento con dati geodetici 173
 - utilizzo viste per l'accesso 120

- comandi
 - db2se 125
- comandi db2se 125
- comando db2trc 152
- comando GET GEOMETRY
 - sintassi 117
- comando gseidx
 - analisi delle statistiche degli indici spaziali 112
 - determinazione delle dimensioni di griglia 111
- comando migrate_v82
 - descrizione 47
- comportamento geodetico
 - ST_Area 344
 - ST_Buffer 354
 - ST_Contains 359
 - ST_Difference 365
 - ST_Distance 370
 - ST_Generalize 387
 - ST_Intersection 403
 - ST_Intersects 405
 - ST_Length 415
 - ST_Perimeter 462
 - ST_SymDifference 484
 - ST_Union 497
 - ST_Within 499
- configurazione database manager
 - parametro, ottimizzazione per le applicazioni spaziali 49
- considerazioni di programmazione
 - programma di esempio di Spatial Extender 133
- conversioni
 - dati spaziali tra sistemi di coordinate 333
 - miglioramento elaborazione delle coordinate 73
- coordinate
 - conversione nel sistema di riferimento spaziale 68
 - conversione per il miglioramento delle prestazioni 73
 - ricerca 319
 - ricerca minime e massime 75
 - sistemi di riferimento spaziali 68
- creazione
 - indici di griglia spaziali 106
 - indici geodetici Voronoi 179

D

- database
 - abilitazione per le operazioni spaziali 54
 - attivazione per le operazioni spaziali panoramica 53
 - configurazione delle applicazioni spaziali 49
 - impostazione delle le applicazioni spaziali 49

- database (*Continua*)
 - migrazione dati spaziali 45
- dati di esempio
 - Spatial Extender 42
- dati di riferimento
 - DB2 Spatial Extender 55
 - impostazione accesso 55
- geocoder 41
- dati geodetici
 - descrizione 4
 - riempimento tabelle con 173
- dati shape, importazione 88
- dati spaziali
 - colonne 81
 - descrizione 3, 4
 - esportazione 87
 - geocoding 93
 - importazione 87
 - richiamo e analisi
 - funzioni 121
 - interfacce 121
 - utilizzo degli indici 122
 - ST_GEOMETRY_COLUMNS 284
 - tipi di dati 81
 - trasferimento dal client al server 509
 - uso 8
- datum
 - geodetico 157, 158
 - nella definizione del sistema di coordinate 220
- datum geodetici
 - descrizione 157
 - sistemi di coordinate 525
 - ST_SPATIAL_REFERENCE_SYSTEMS 291
- DB2 Geodetic Extender
 - funzioni spaziali supportate 206
- DE_HDN_SRS_1004
 - sistema di riferimento spaziale 71
- DEFAULT_SRS
 - sistema di riferimento spaziale 71
- densità di popolazione mondiale
 - struttura di celle Voronoi 176
- distanza
 - con una geodesica 160
 - funzione ST_Distance 370

E

- ellissoidi
 - Geodetic Extender 220
- emisferi
 - poligoni che li rappresentano 197
- equatore 159
- esempi
 - Spatial Extender 135
- esportazione dati
 - dati
 - file di trasferimento SDE 92
 - file shape 91

Indice

estensione spaziale
 definizione 68
estensioni
 creazione di un sistema di riferimento
 spaziale utilizzandole 75
estensioni geografiche, definizione 75

F

fattori, conversione
 coordinate 73
fattori di scala
 calcolo per il nuovo sistema di
 riferimento spaziale 75
 panoramica 73
file di intestazione, incluso DB2 Spatial
 Extender 133
file di intestazione h 133
file di trasferimento SDE
 esportazione dati in 92
 importazione dati da 90
file shape
 esportazione dati in 91
formati dati
 GML (Geography Markup
 Language) 522
 rappresentazione shape 522
 rappresentazione WKB (well-known
 binary) 520
 rappresentazione WKT (well-known
 text) 515
forme geometriche
 dati spaziali 8
 generazione nuova
 conversione da una all'altra 325
 forme modificate 331
 in base alle misure esistenti 330
 nuove configurazioni di
 spazio 326
 panoramica 325
 una da molte 330
 panoramica 11
 proprietà
 panoramica 13
 Vedere anche "Funzioni spaziali,
 proprietà delle forme
 geometriche" 318
 trasferimento dati client-server 509
formule utilizzate nel geocoding 73
funzione aggregata
 colonne spaziali 341, 507
funzioni
 spaziali
 conversione formato nello scambio
 dati 295
 panoramica 295
funzioni aggregate di unione 507
funzioni di confronto
 forme geometriche identiche 315
 intersezioni tra forme
 geometriche 309, 317
 involuppi forme geometriche 315
 panoramica 304
 relazioni contenitore 306
 stringa matrice modello DE-9IM 318
funzioni di costruzione
 panoramica 296

funzioni di costruzione (*Continua*)
 Rappresentazione GML (Geography
 Markup Language) 303
 Rappresentazione shape ESRI 302
 rappresentazione WKB (well-known
 binary) 301
 rappresentazione WKT (well-known
 text) 300
funzioni geografiche
 descrizione 3
 rappresentate da dati 4
funzioni spaziali
 aggregazione MBR 341
 Aggregazione unione 507
 che usano indici geodetici
 Voronoi 175, 180
 confronto delle forme geometriche
 forme geometriche identiche 315
 intersezioni 309, 317
 involuppi forme geometriche 315
 panoramica 304
 relazioni contenitore 306
 stringa matrice modello
 DE-9IM 318
 considerazioni 335
 conversione di dati tra i sistemi di
 coordinate 333
 conversione formato nello scambio
 dati
 panoramica 296
 Rappresentazione GML
 (Geography Markup
 Language) 303
 Rappresentazione shape ESRI 302
 rappresentazione WKB
 (well-known binary) 301
 rappresentazione WKT
 (well-known text) 300
 conversione forme geometriche 295
 differenza geodetica nel
 comportamento 206
 EnvelopesIntersect 339
 esempi 121
 generazione nuove forme geometriche
 conversione da una all'altra 325
 forme modificate 331
 in base alle misure esistenti 330
 nuove configurazioni di
 spazio 326
 panoramica 325
 una da molte 330
 informazioni sull'indice 333
 informazioni sulla distanza 332
 obsolete 561
 panoramica 295
 proprietà forme geometriche 318
 forme geometriche in una forma
 geometrica 321
 informazioni sul contorno 322
 informazioni sul tipo di dati 318
 informazioni sulla
 configurazione 324
 informazioni sulle coordinate e le
 misure 319
 informazioni sulle dimensioni 323
 sistema di riferimento
 spaziale 324

funzioni spaziali (*Continua*)
 ST_AppendPoint 343
 ST_Area 344
 ST_AsBinary 348
 ST_AsGML 349
 ST_AsShape 350
 ST_AsText 351
 ST_Boundary 352
 ST_Buffer 354
 ST_Centroid 357
 ST_ChangePoint 358
 ST_Contains 359
 ST_ConvexHull 361
 ST_CoordDim 363
 ST_Crosses 364
 ST_Difference 365
 ST_Dimension 367
 ST_Disjoint 368
 ST_Distance 370
 ST_Edge_GC_USA 373
 ST_Endpoint 377
 ST_Envelope 378
 ST_EnvIntersects 379
 ST_EqualCoordsys 380
 ST_Equals 381
 ST_EqualsSRS 383
 ST_ExteriorRing 384
 ST_FindMeasure
 ST_LocateAlong 385
 ST_Generalize 387
 ST_GeomCollection 388
 ST_GeomCollFromTxt 390
 ST_GeomCollFromWKB 392
 ST_Geometry 393
 ST_GeometryN 395
 ST_GeometryType 396
 ST_GeomFromText 397
 ST_GeomFromWKB 398
 ST_GetIndexParms 400
 ST_InteriorRingN 402
 ST_Intersection 403
 ST_Intersects 405
 ST_Is3d 407
 ST_IsClosed 408
 ST_IsEmpty 409
 ST_IsMeasured 410
 ST_IsRing 411
 ST_IsSimple 412
 ST_IsValid 414
 ST_Length 415
 ST_LineFromText 417
 ST_LineFromWKB 418
 ST_LineString 419
 ST_LineStringN 420
 ST_LocateAlong
 ST_FindMeasure 385
 ST_LocateBetween
 ST_MeasureBetween 432
 ST_M 422
 ST_MaxM 423
 ST_MaxX 425
 ST_MaxY 426
 ST_MaxZ 428
 ST_MBR 429
 ST_MBRIntersects 430
 ST_MeasureBetween
 ST_LocateBetween 432

funzioni spaziali (*Continua*)

ST_MidPoint 434
 ST_MinM 435
 ST_MinX 436
 ST_MinY 438
 ST_MinZ 439
 ST_MLineFromText 440
 ST_MLineFromWKB 442
 ST_MPointFromText 444
 ST_MPointFromWKB 445
 ST_MPolyFromText 446
 ST_MPolyFromWKB 448
 ST_MultiLineString 450
 ST_MultiPoint 451
 ST_MultiPolygon 453
 ST_NumGeometries 455
 ST_NumInteriorRing 456
 ST_NumLineStrings 457
 ST_NumPoints 458
 ST_NumPolygons 459
 ST_Overlaps 460
 ST_Perimeter 462
 ST_PerpPoints 463
 ST_Point 465
 ST_PointFromText 468
 ST_PointFromWKB 469
 ST_PointN 471
 ST_PointOnSurface 472
 ST_PolyFromText 473
 ST_PolyFromWKB 474
 ST_Polygon 475
 ST_PolygonN 478
 ST_Relate 479
 ST_RemovePoint 480
 ST_SRID
 ST_SrsId 481
 ST_SrsID
 ST_SRID 481
 ST_SrsName 482
 ST_StartPoint 483
 ST_SymDifference 484
 ST_ToGeomColl 487
 ST_ToLineString 488
 ST_ToMultiLine 489
 ST_ToMultiPoint 490
 ST_ToMultiPolygon 491
 ST_ToPoint 492
 ST_ToPolygon 493
 ST_Touches 494
 ST_Transform 495
 ST_Union 497
 ST_Within 499
 ST_WKBToSQL 501
 ST_WKTToSQL 502
 ST_X 503
 ST_Y 504
 ST_Z 506
 tipi di dati associati 335
 utilizzo per l'impostazione degli indici spaziali 122

G

GCS_NORTH_AMERICAN_1927
 sistema di coordinate 71
 GCS_NORTH_AMERICAN_1983
 sistema di coordinate 71

GCS_WGS_1984
 sistema di coordinate 71
 GCSW_DEUTSCHE_HAUPTDREIECKSNETZ
 sistema di coordinate 71
 geocoder
 dati di riferimento 41
 esecuzione in modalità batch 97
 panoramica 93
 registrazione 56
 vista del catalogo ST_GEOCODER_PARAMETERS 285
 vista del catalogo ST_GEOCODERS 287
 vista del catalogo ST_GEOCODING_PARAMETERS 288
 vista del catalogo ST_SIZINGS 290
 geocoding
 impostazione 94
 modalità batch 97
 panoramica 93
 geocoding automatico 93, 96
 geocoding in batch 93
 Geodesia 157
 geodesico
 definizione 160
 esempio 197
 Geodetic Extender
 attributi ST_Geometry 197
 descrizione 157
 differenze 197
 ellissoidi 220
 impostazione 165
 procedure memorizzate spaziali supportate 211
 quando utilizzarlo 158
 viste del catalogo spaziale supportate 211
 geodetica, latitudine
 definizione di 159
 geodetica, longitudine
 definizione di 159
 geodetico, datum 158
 geografica, carta
 esempi forniti con il prodotto 42
 GEOMETRY_COLUMNS, vista del catalogo spaziale obsoleta 557
 GeometryFromShape, funzione spaziale obsoleta 561
 GML (Geographic Markup Language), formato dati 522
 gradi
 latitudine e longitudine 159
 gruppi di trasformazione
 panoramica 509
 gse_disable_autogc, procedura memorizzata spaziale obsoleta 535
 gse_disable_sref, procedura memorizzata spaziale obsoleta 535
 gse_enable_autogc, procedura memorizzata spaziale obsoleta 535
 gse_enable_db, procedura memorizzata spaziale obsoleta 535
 gse_enable_idx, procedura memorizzata spaziale obsoleta 535

gse_enable_sref, procedura memorizzata spaziale obsoleta 535
 gse_export_shape 254
 gse_import_shape, procedura memorizzata spaziale obsoleta 535
 gse_register_gc, procedura memorizzata spaziale obsoleta 535
 gse_register_layer, procedura memorizzata spaziale obsoleta 535
 gse_run_gc, procedura memorizzata spaziale obsoleta 535
 gse_unregist_gc, procedura memorizzata spaziale obsoleta 535
 gse_unregist_layer, procedura memorizzata spaziale obsoleta 535

H

HP-UX
 installazione
 DB2 Spatial Extender 31

I

ID del sistema di riferimento spaziale (SRID)
 per geodetica 157, 158
 ID del sistema di riferimento spaziale geodetico
 ST_create_srs 238
 importazione
 dati di trasferimento SDE 90
 dati shape 88
 impostazione
 DB2 Spatial Extender 25
 impostazioni
 geocoding automatico 96
 operazioni di geocoding 94
 Index Advisor
 analisi delle statistiche degli indici spaziali 112
 comando GET GEOMETRY da richiamare 117
 determinazione delle dimensioni di griglia 111
 quando utilizzarlo 102
 scopo 100, 110
 indice di griglia spaziale
 analisi delle statistiche degli indici spaziali 112
 comando Index Advisor 117
 determinazione delle dimensioni di griglia 111
 funzioni spaziali che l'utilizzano 108
 istruzione CREATE INDEX 108
 istruzioni SQL che l'utilizzano 108
 indici
 analisi delle statistiche degli indici spaziali 112
 comando Index Advisor 117
 creazione di un geodetico Voronoi 179
 creazione indice di griglia spaziale 106
 determinazione delle dimensioni di griglia 111

Indice

indici (*Continua*)
 indice di griglia spaziale 100
 istruzione CREATE INDEX per geodetici Voronoi 180
 istruzione CREATE INDEX per la griglia spaziale 108
 struttura cella geodetica Voronoi 177

indici di griglia
 creazione 106
 ottimizzazione 110
 panoramica 100

indici di griglia spaziali
 creazione 106
 impostazione 122
 livelli di griglia e dimensioni 100, 102
 paragonati agli indici geodetici Voronoi 99

indici geodetici Voronoi
 creazione 179
 funzioni che li utilizzano 175
 impostazione 122
 istruzione CREATE INDEX 180
 paragonati agli indici di griglia spaziali 99
 selezione di una struttura Voronoi alternativa 177

indici spaziali
 geodetici Voronoi 175
 tipi di 99

informazioni sul tipo di dati, ricerca 318

informazioni sull'indice per le forme geometriche 333

informazioni sulla distanza per le forme geometriche 332

informazioni sulle misure, ricerca 319

installazione
 DB2 Spatial Extender
 AIX 29
 HP-UX 31
 Linux e Linux 390 35
 requisiti hardware e software 26
 Solaris Operating Environment 33
 verifica 39
 Windows 27
 istanze, creazione 37
 Spatial Extender 25

interfacce
 creazione di un sistema di riferimento spaziale 75
 DB2 Spatial Extender 17

intero pianeta
 rappresentazione 197

Is3d, funzione spaziale obsoleta 561

IsMeasured, funzione spaziale obsoleta 561

istanze
 creazione 37

istruzione CREATE INDEX
 indice di griglia spaziale 108
 indice geodetici Voronoi 180

istruzioni SQL
 che usano indici geodetici Voronoi 180

L

latitudine geodetica 159

licenza
 per Geodetic Extender 165

linee 11

LineFromShape, funzione spaziale obsoleta 561

LocateAlong, funzione spaziale obsoleta 561

LocateBetween, funzione spaziale obsoleta 561

longitudine geodetica 159

M

M, funzione spaziale obsoleta 561

meridiani primi
 sistemi di coordinate 525

meridiano 159

messaggi
 Centro di controllo 151
 funzioni 147
 informazioni sulla forma 149
 informazioni sulla migrazione 149
 Spatial Extender
 CLP 149
 parte di 143
 procedure memorizzate 145

messaggi di funzione 147

migrazione
 Spatial Extender 45, 47

minimum bounding circle (MBC)
 attributi ST_Geometry 197
 definizione 175
 funzioni spaziali risultanti 206

minimum bounding rectangle (MBR)
 definizione 13

Minimum bounding rectangle (MBR)
 uso negli indici di griglia spaziali 100

MLineFromShape, funzione spaziale obsoleta 561

moltiplicatori per il miglioramento delle prestazioni
 elaborazione delle coordinate 73

MPointFromShape, funzione spaziale obsoleta 561

MPolyFromShape, funzione spaziale obsoleta 561

multilinee, raccolta omogenea in Spatial Extender 11

multipoligoni, raccolta omogenea in Spatial Extender 11

multipunti, raccolta omogenea in Spatial Extender 11

N

NAD27_SRS_1002
 sistema di riferimento spaziale 71

NAD83_SRS_1
 sistema di riferimento spaziale 71

P

parametri di configurazione
 applicazioni spaziali
 ottimizzazione 49
 valori 49

parametri di configurazione del database
 applicazioni spaziali
 ottimizzazione 49
 parametro
 APP_CTL_HEAP_SZ 49
 parametro APPLHEAPSZ 49
 parametro LOGFILSZ 49
 parametro LOGPRIMARY 49
 parametro LOGSECOND 49
 parametro APP_CTL_HEAP_SZ,
 ottimizzazione 49

parametro di configurazione
 APPLHEAPSZ
 ottimizzazione 49

parametro di configurazione dimensione heap controllo dell'applicazione 49

parametro di configurazione
 LOGFILSZ 49

parametro di configurazione
 LOGPRIMARY 49

parametro di configurazione
 LOGSECOND
 ottimizzazione 49

parametro dimensione heap dell'applicazione (APPLHEAPSZ) 49

PointFromShape, funzione spaziale obsoleta 561

poli
 poligoni inclusi 197

poligoni
 definizione di regioni geodetiche 161
 tipo di forma geometrica 11

poligoni geodetici 161

prestazioni
 conversioni coordinate-dati 73

primo meridiano 159

procedura memorizzata
 gse_disable_autogc 244

procedura memorizzata
 gse_disable_db 246

procedura memorizzata
 gse_disable_sref 249

procedura memorizzata
 gse_enable_autogc 250

procedura memorizzata
 gse_enable_db 252

procedura memorizzata
 gse_enable_sref 238

procedura memorizzata
 GSE_export_sde 226

procedura memorizzata
 gse_import_sde 228

procedura memorizzata
 GSE_import_sde 228

procedura memorizzata
 gse_import_shape 257

procedura memorizzata
 gse_register_gc 265

procedura memorizzata
 gse_register_layer 270

procedura memorizzata gse_run_gc 273

procedura memorizzata
 gse_unregist_gc 280
 procedura memorizzata
 gse_unregist_layer 281
 procedura memorizzata
 ST_alter_coordsys 230
 procedura memorizzata
 ST_create_coordsys 236
 procedura memorizzata
 ST_disable_db 246
 procedura memorizzata
 ST_drop_coordsys 248
 procedura memorizzata
 ST_enable_autogeocoding 250
 procedura memorizzata
 ST_enable_db 252
 procedura memorizzata
 ST_export_shape 254
 procedura memorizzata
 ST_import_shape 257
 procedura memorizzata
 ST_register_geocoder 265
 procedura memorizzata
 ST_register_spatial_column 270
 procedura memorizzata
 ST_remove_geocoding_setup 271
 procedura memorizzata
 ST_run_geocoding 273
 procedura memorizzata
 ST_setup_geocoding 276
 procedura memorizzata
 ST_unregister_geocoder 280
 procedura memorizzata
 ST_unregister_spatial_column 281
 procedure memorizzate
 GSE_export_sde 226
 GSE_import_sde 228
 problemi 145
 richiamo
 applicazioni spaziali 133
 richiamo dalle applicazioni
 spaziali 134
 ST_alter_coordsys 230
 ST_alter_srs 232
 ST_create_coordsys 236
 ST_create_srs 238
 ST_disable_autogeocoding 244
 ST_disable_db 246
 ST_drop_coordsys 248
 ST_drop_srs 249
 ST_enable_autogeocoding 250
 ST_enable_db 252
 ST_export_shape 254
 ST_import_shape 257
 ST_register_geocoder 265
 ST_register_spatial_column 270
 ST_remove_geocoding_setup 271
 ST_run_geocoding 273
 ST_setup_geocoding 276
 ST_unregister_geocoder 280
 ST_unregister_spatial_column 281
 procedure memorizzate spaziali
 obsolete 535
 supportate da Geodetic Extender 211
 proiezioni area equivalente 65
 proiezioni azimutali 65

proiezioni carte geografiche
 sistemi di coordinate 525
 proiezioni conformi 65
 proiezioni direzione reale 65
 proiezioni equidistanti 65
 proprietà forme geometriche
 funzioni spaziali per 318
 forme geometriche in una forma
 geometrica 321
 informazioni sul contorno 322
 informazioni sul tipo di dati 318
 informazioni sulla
 configurazione 324
 informazioni sulle coordinate e le
 misure 319
 informazioni sulle dimensioni 323
 sistema di riferimento
 spaziale 324
 panoramica 13
 punti 11

Q

query
 funzioni spaziali da eseguire 121
 indici spaziali, utilizzo 122
 spaziali, interfacce da inoltrare 121

R

rappresentazione shape, formato
 dati 522
 rappresentazione WKB (well-known
 binary), formati dati 520
 rappresentazione WKT (well-known text),
 formato dati 515
 regioni geodetiche
 descrizione 161
 registrazione
 colonne spaziali 85
 geocoder 56
 registrazione eventi per l'individuazione
 dei problemi 152
 registrazione notifiche di gestione 153
 registrazioni
 diagnostica 153
 regolazione degli indici di griglia
 utilizzo di Index Advisor 111
 regolazione degli indici di griglia spaziali
 con Index Advisor 110
 requisiti del sistema
 per Geodetic Extender 165
 requisiti hardware
 Spatial Extender 26
 requisiti software
 Spatial Extender 26
 risoluzione dei problemi
 funzioni 147
 messaggi di informazioni forma 149
 messaggi di migrazione 149
 registrazione notifiche di
 gestione 153
 Spatial Extender
 mediante runGseDemo 40
 messaggi 143
 procedure memorizzate 145

risoluzione dei problemi (*Continua*)
 Spatial Extender (*Continua*)
 programma di esempio 40
 traccia 152

S

scenari
 impostazione di Spatial Extender 17
 sferoidi
 definizione 158
 nella definizione del sistema di
 coordinate 220
 sistemi di coordinate 525
 ShapeToSQL, funzione spaziale
 obsoleta 561
 sistema di coordinate geografiche 59
 sistema di coordinate proiettate 59
 sistema di riferimento di coordinate
 latitudine e longitudine 157
 sistema di riferimento spaziale geodetico
 descrizione 68
 sistemi di coordinate
 creazione 66
 panoramica 59
 selezione 66
 supportati 525
 vista del catalogo ST_COORDINATE_
 SYSTEMS 283
 vista del catalogo ST_SPATIAL_
 REFERENCE_SYSTEMS 291
 sistemi di coordinate proiettate 65
 sistemi di riferimento spaziali
 creazione 75, 238
 descrizione 68
 fornito con DB2 Spatial Extender 71
 predefiniti 69
 sistemi di riferimento spaziali
 geodetici 157
 sistemi di riferimento spaziali
 predefiniti 69
 Solaris Operating Environment
 installazione
 DB2 Spatial Extender 33
 Spatial Extender
 abilitazione 54
 dati di riferimento 55
 impostazione accesso 55
 installazione 35
 quando utilizzarlo 158
 sistemi di riferimento spaziali forniti
 con 71
 SPATIAL_GEOCODER, vista del catalogo
 spaziale obsoleta 557
 SPATIAL_REF_SYS, vista del catalogo
 spaziale obsoleta 557
 ST_alter_srs 232
 ST_COORDINATE_SYSTEMS 283
 ST_create_srs 238
 ST_disable_autogeocoding 244
 ST_Distance 370
 ST_drop_srs 249
 ST_GEOCODER_PARAMETERS 285
 ST_GEOCODERS 287
 ST_GEOCODING 287
 ST_GEOCODING_PARAMETERS 288
 ST_GEOMETRY_COLUMNS 284

Indice

ST_GeomFromText, funzione spaziale
obsoleta 561

ST_GeomFromWKB, funzione spaziale
obsoleta 561

ST_LineFromText, funzione spaziale
obsoleta 561

ST_LineFromWKB, funzione spaziale
obsoleta 561

ST_MLineFromText, funzione spaziale
obsoleta 561

ST_MLineFromWKB, funzione spaziale
obsoleta 561

ST_MPointFromText, funzione spaziale
obsoleta 561

ST_MPointFromWKB, funzione spaziale
obsoleta 561

ST_MPolyFromText, funzione spaziale
obsoleta 561

ST_MPolyFromWKB, funzione spaziale
obsoleta 561

ST_OrderingEquals, funzione spaziale
obsoleta 561

ST_Point, funzione spaziale obsoleta 561

ST_PointFromText, funzione spaziale
obsoleta 561

ST_PolyFromText, funzione spaziale
obsoleta 561

ST_PolyFromWKB, funzione spaziale
obsoleta 561

ST_SIZINGS 290

ST_SPATIAL_
REFERENCE_SYSTEMS 291

ST_SymmetricDiff, funzione spaziale
obsoleta 561

ST_Transform, funzione spaziale
obsoleta 561

ST_UNITS_OF_MEASURE 293

strutture di celle Voronoi
descrizione 176
selezione di una alternativa per
l'indice 177

T

tabelle
colonne spaziali 83
importazione dati shape 88

tassellazione Voronoi 176

U

unità di misura angolari
sistemi di coordinate 525

unità di misura lineari
sistemi di coordinate 525

unità per i valori di scarto e fattori di
scala 73

V

valori di scarto
calcolo per il nuovo sistema di
riferimento spaziale 75
panoramica 73

verifica
installazione di Spatial Extender 39

vista del catalogo spaziale
COORD_REF_SYS, obsoleta 557

vista del catalogo ST_UNITS_OF_
MEASURE 293

viste
DB2 Spatial Extender
accesso alle colonne spaziali 120

viste del catalogo
ST_COORDINATE_SYSTEMS 283
ST_GEOCODER_PARAMETERS 285
ST_GEOCODERS 287
ST_GEOCODING 287
ST_GEOCODING_
PARAMETERS 288
ST_GEOMETRY_COLUMNS 284
ST_SIZINGS 290
ST_SPATIAL_
REFERENCE_SYSTEMS 291
ST_UNITS_OF_MEASURE 293

viste del catalogo spaziale
supportate da Geodetic Extender 211

viste del catalogo spaziale, obsolete
COORD_REF_SYS 557
GEOMETRY_COLUMNS 557
SPATIAL_GEOCODER 557
SPATIAL_REF_SYS 557

W

WGS84_SRS_1003
sistema di riferimento spaziale 71

Windows
installazione
DB2 Spatial Extender 27

Z

Z, funzione spaziale obsoleta 561

Come ottenere ulteriori informazioni dalla IBM

Negli Stati Uniti, per contattare la IBM chiamare uno dei seguenti numeri:

- 1-800-IBM-SERV (1-800-426-7378) per contattare l'assistenza clienti
-
- 1-800-IBM-4YOU (426-4968) per contattare la sezione DB2 Marketing and Sales

In Canada, per contattare la IBM chiamare uno dei seguenti numeri:

- 1-800-IBM-SERV (1-800-426-7378) per contattare l'assistenza clienti
- 1-800-465-9600 per informazioni sui servizi disponibili
- 1-800-IBM-4YOU (1-800-426-4968) per contattare la sezione DB2 Marketing and Sales

Per contattare un ufficio IBM nel proprio paese, visitare il sito Web Directory of Worldwide Contacts della IBM <http://www.ibm.com/planetwide>

Informazioni sul prodotto

Informazioni relative ai prodotti DB2 Universal Database sono disponibili telefonicamente o sul Web all'indirizzo <http://www.ibm.com/software/data/db2/udb>

Questo sito contiene le informazioni aggiornate sulla libreria tecnica, su come ordinare le pubblicazioni, sul download dei prodotti, su newsgroup, FixPak, novità e collegamenti alle risorse web.

- Per ordinare prodotti e ottenere informazioni generali, chiamare il numero 1-800-IBM-CALL (1-800-426-2255).
-

Per informazioni su come contattare l'IBM al di fuori degli Stati Uniti, visitare la pagina Web all'indirizzo www.ibm.com/planetwide



Stampato in Italia

SC13-3223-01



Spine information:



IBM® DB2® Spatial Extender e
Geodetic Extender

DB2 Spatial Extender e Geodetic Extender - Guida di
riferimento per l'utente

Versione 8.2