

IBM DB2 Universal Database



Lernprogramm für Visual Explain

Version 8

IBM DB2 Universal Database



Lernprogramm für Visual Explain

Version 8

Hinweis:

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter *Bemerkungen* gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Universal Database Visual Explain Tutorial, Version 8,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2002
© Copyright IBM Deutschland Informationssysteme GmbH 2002

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
September 2002

Inhaltsverzeichnis

Zu diesem Lernprogramm	v
Umgebungsspezifische Informationen	vi

Lerneinheit 1. Erstellen von EXPLAIN-Momentaufnahmen.	1
Erstellen von EXPLAIN-Tabellen	1
Verwenden von EXPLAIN-Momentaufnahmen	2
Erstellen von EXPLAIN-Momentaufnahmen für dynamische SQL-Anweisungen.	4
Erstellen von EXPLAIN-Momentaufnahmen für statische SQL-Anweisungen	5
Weitere Schritte	5

Lerneinheit 2. Anzeigen und Verwenden eines Zugriffspdiagramms	7
Anzeigen eines Zugriffspdiagramms durch Auswählen aus einer Liste von zuvor mit EXPLAIN bearbeiteten SQL-Anweisungen.	7
Verstehen der Symbole in einem Zugriffspdiagramm	8
Verwenden der Zoomleiste zum Vergrößern von Teilen eines Diagramms	8
Anzeigen weiterer Details zu den Objekten in einem Diagramm.	9
Anzeigen von Statistikdaten für Tabellen, Indizes und Tabellenfunktionen	9
Anzeigen von Details zu Operatoren in einem Diagramm	10
Anzeigen von Statistikdaten für Funktionen	10
Anzeigen von Statistikdaten für Tabellenbereiche	11
Anzeigen von Statistikdaten für Spalten in einer SQL-Anweisung.	11
Anzeigen von Informationen zu Konfigurationsparametern und Bindeoptionen	11
Ändern der Darstellung eines Diagramms	11
Weitere Schritte	12

Lerneinheit 3. Verbessern eines Zugriffspdiagramms in einer Umgebung mit Datenbanken mit einer Partition	13
Arbeiten mit Zugriffspdiagrammen	13

Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten.	14
Erfassen aktueller Statistikdaten für die Tabellen und Indizes mit Hilfe des Befehls 'runstats'	17
Erstellen von Indizes für Spalten, die verwendet werden, um Tabellen in einer Abfrage zu verknüpfen	22
Erstellen zusätzlicher Indizes für Tabellenspalten.	27
Weitere Schritte	29

Lerneinheit 4. Verbessern eines Zugriffspdiagramms in einer Umgebung mit partitionierten Datenbanken	31
Arbeiten mit Zugriffspdiagrammen	31
Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten.	32
Erfassen aktueller Statistikdaten für die Tabellen und Indizes mit Hilfe des Befehls 'runstats'	35
Erstellen von Indizes für Spalten, die verwendet werden, um Tabellen in einer Abfrage zu verknüpfen	40
Erstellen zusätzlicher Indizes für Tabellenspalten.	44
Weitere Schritte	46

Anhang A. Visual Explain-Konzepte	47
Abfrageoptimierungsklasse	47
Aufwand	48
Behälter	48
Clustering.	48
Cursorblockung.	49
Dynamisches SQL	49
EXPLAIN-Momentaufnahme	50
Mit EXPLAIN bearbeitbare Anweisung	51
Mit EXPLAIN bearbeitete Anweisung	51
Operand	51
Operator	51
CMPEXP	53
DELETE	53
EISCAN	53
FETCH.	54
FILTER.	54

GENROW.	55	DELETE	75
GRPBY.	55	EISCAN	75
HSJOIN	56	FETCH.	76
INSERT	57	FILTER.	76
IXAND	57	GENROW.	77
IXSCAN	58	GRPBY.	77
MSJOIN	59	HSJOIN	78
NLJOIN	60	INSERT	79
PIPE	61	IXAND	79
RETURN	61	IXSCAN	80
RIDSCN	61	MSJOIN	81
RQUERY	62	NLJOIN	81
SORT	62	PIPE	82
TBSCAN	63	RETURN	82
TEMP	64	RIDSCN	83
TQUEUE	65	RQUERY	83
UNION	65	SORT	83
UNIQUE	65	TBSCAN	85
UPDATE	66	TEMP	86
Optimierungsprogramm	66	TQUEUE	86
Paket	66	UNION	86
Selektivität der Vergleichselemente	67	UNIQUE	87
Statisches SQL	67	UPDATE	87
Sternverknüpfungen	68	Anhang C. DB2-Konzepte	89
Tabellenbereich	69	Datenbanken.	89
Vergleichselement	69	Schemata	89
Visual Explain	70	Tabellen	90
Vom Betriebssystem verwaltete Tabellen- bereiche (SMS)	71	Anhang D. Bemerkungen	91
Vom Datenbankmanager verwalteter Tabellen- bereich (DMS)	71	Marken	94
Zugriffsplan	72	Index	97
Zugriffsplandiagramm	73	Kontaktaufnahme mit IBM.	99
Zugriffsplandiagrammknoten	74	Produktinformationen.	99
Anhang B. Alphabetische Liste der Visual Explain-Operatoren	75		
CMPEXP	75		

Zu diesem Lernprogramm

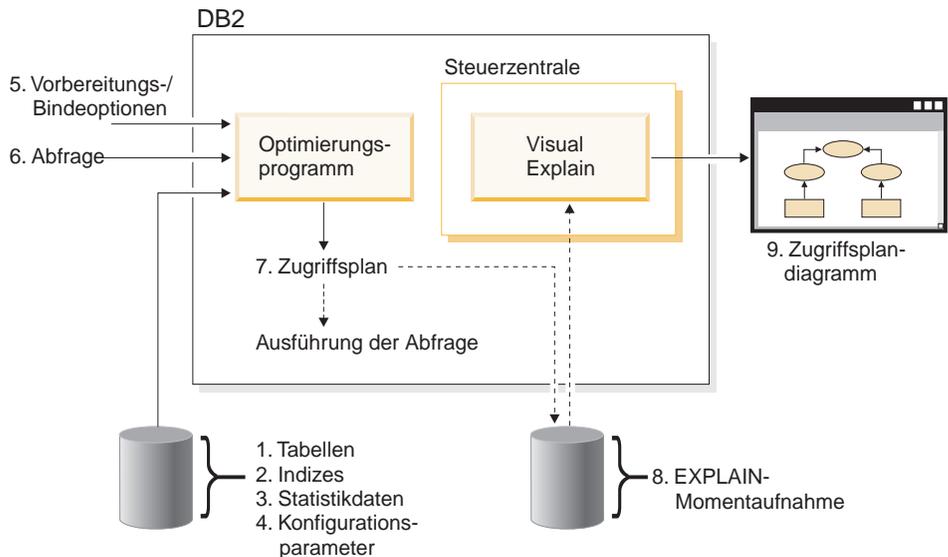
Dieses Lernprogramm führt Sie in die Merkmale von DB2 Visual Explain ein. Wenn Sie die Lerneinheiten in diesem Lernprogramm durcharbeiten, erfahren Sie, wie Sie mit Visual Explain den Zugriffsplan für mit EXPLAIN bearbeitete SQL-Anweisungen als Diagramm anzeigen können. Darüber hinaus erfahren Sie, wie Sie die Informationen aus diesem Diagramm verwenden können, um Ihre SQL-Abfragen so zu optimieren, dass die Leistung erhöht wird.

Mit Hilfe des Optimierungsprogramms untersucht DB2 Ihre SQL-Abfragen und ermittelt die beste Zugriffsmöglichkeit auf Ihre Daten. Dieser Pfad zu den Daten wird Zugriffsplan genannt. DB2 ermöglicht es Ihnen, die Aktionen des Optimierungsprogramms anzuzeigen, indem Sie den Zugriffsplan anzeigen können, der zur Ausführung einer bestimmten SQL-Abfrage ausgewählt wurde. Sie können den Zugriffsplan mit Visual Explain als Diagramm anzeigen. Das Diagramm ist eine visuelle Darstellung der Datenbankobjekte, die an einer Abfrage beteiligt sind (z. B. Tabellen und Indizes). Darüber hinaus enthält das Diagramm die Operationen, die an diesen Objekten ausgeführt wurden (z. B. Durchsuchungen und Sortierungen), und zeigt den Datenfluss an.

Sie können den Zugriff einer Abfrage auf Daten verbessern, indem Sie eine oder alle der folgenden Optimierungsaktivitäten ausführen:

1. Optimieren Sie Ihren Tabellenentwurf, und reorganisieren Sie die Tabellendaten.
2. Erstellen Sie die entsprechenden Indizes.
3. Verwenden Sie den Befehl **runstats**, um dem Optimierungsprogramm die aktuellen Statistikdaten zur Verfügung zu stellen.
4. Wählen Sie die entsprechenden Konfigurationsparameter aus.
5. Wählen Sie die entsprechenden Bindeoptionen aus.
6. Entwerfen Sie Abfragen, mit denen nur erforderliche Daten abgerufen werden.
7. Arbeiten Sie mit einem Zugriffsplan.
8. Erstellen Sie EXPLAIN-Momentaufnahmen.
9. Verwenden Sie ein Zugriffsplandiagramm, um einen Zugriffsplan zu verbessern.

Diese leistungsbezogenen Aktivitäten entsprechen den Aktivitäten in der folgenden Darstellung. (Gestrichelte Linien geben die Aktionen an, die für Visual Explain erforderlich sind.)



Dieses Lernprogramm enthält Lerneinheiten zu folgenden Themen:

- Erstellen von EXPLAIN-Momentaufnahmen. Diese sind für die Anzeige von Zugriffsplandiagrammen erforderlich.
- Anzeigen und Ändern eines Zugriffsplandiagramms.
- Ausführen von Optimierungsaktivitäten und Durchführen von Untersuchungen, wie diese Aktivitäten Ihren Zugriffsplan verbessern.

Anmerkung: Die Leistungsverbesserung ist in eine Lerneinheit für Umgebungen mit Datenbanken mit einer Partition und in eine Lerneinheit für Umgebungen mit partitionierten Datenbanken unterteilt.

Sie verwenden in diesen Lerneinheiten die zum Lieferumfang von DB2 gehörende Beispieldatenbank SAMPLE. Wenn Sie die Beispieldatenbank noch nicht erstellt haben, finden Sie Informationen dazu im Handbuch *Systemverwaltung*.

Umgebungsspezifische Informationen



Informationen, die mit diesem Symbol gekennzeichnet sind, beziehen sich nur auf Umgebungen mit Datenbanken mit einer Partition.



Informationen, die mit diesem Symbol gekennzeichnet sind, beziehen sich nur auf Umgebungen mit partitionierten Datenbanken.

Lerneinheit 1. Erstellen von EXPLAIN-Momentaufnahmen

In dieser Lerneinheit erstellen Sie EXPLAIN-Momentaufnahmen. Mit der SQL-EXPLAIN-Einrichtung werden Informationen zu der Umgebung erfasst, in der eine statische oder dynamische SQL-Anweisung kompiliert wird. Anhand der erfassten Informationen können Sie die Struktur und die mögliche Ausführungsleistung Ihrer SQL-Anweisungen besser verstehen. Eine EXPLAIN-Momentaufnahme besteht aus komprimierten Informationen, die erfasst werden, wenn eine SQL-Anweisung mit EXPLAIN bearbeitet wird. Sie wird als großes Binärobjekt (BLOB) in der Tabelle EXPLAIN_STATEMENT gespeichert und enthält die folgenden Informationen:

- Die interne Darstellung des Zugriffsplans, einschließlich seiner Operatoren und der Tabellen und Indizes, auf die zugegriffen wird.
- Die Entscheidungsbedingungen, die vom Optimierungsprogramm verwendet werden, einschließlich der Statistikdaten für Datenbankobjekte und des kumulativen Aufwands für jede Operation.

Visual Explain benötigt die Informationen, die in einer EXPLAIN-Momentaufnahme enthalten sind, um ein Zugriffsplandiagramm anzuzeigen.

Erstellen von EXPLAIN-Tabellen

Wenn Sie EXPLAIN-Momentaufnahmen erstellen möchten, müssen Sie sicherstellen, dass die folgenden EXPLAIN-Tabellen für Ihre Benutzer-ID vorhanden sind:

- EXPLAIN_INSTANCE
- EXPLAIN_STATEMENT

Verwenden Sie den DB2-Befehl **list tables**, um zu überprüfen, ob diese Tabellen vorhanden sind. Wenn diese Tabellen nicht vorhanden sind, müssen Sie sie unter Beachtung der folgenden Anweisungen erstellen:

1. Wenn DB2 noch nicht gestartet wurde, setzen Sie den Befehl **db2start** ab.
2. Stellen Sie über die Eingabeaufforderung des DB2-Befehlszeilenprozessors eine Verbindung zu der Datenbank her, die Sie verwenden möchten. Stellen Sie für diese Lerneinheit eine Verbindung zur Beispieldatenbank SAMPLE über den Befehl **connect to sample** her.
3. Erstellen Sie die EXPLAIN-Tabellen, indem Sie die Beispielbefehlsdatei in der Datei EXPLAIN.DDL verwenden. Diese Datei befindet sich im Verzeichnis `sqlib\misc`. Wechseln Sie in dieses Verzeichnis, und setzen Sie den Befehl **db2 -tf EXPLAIN.DDL** ab, um die Befehlsdatei auszuführen.

Diese Befehlsdatei erstellt EXPLAIN-Tabellen, die als Präfix die ID des verbundenen Benutzers erhalten. Diese Benutzer-ID muss das Zugriffsrecht CREATETAB für die Datenbank oder die Berechtigung SYSADM oder DBADM haben.

Verwenden von EXPLAIN-Momentaufnahmen

Es gibt vier Beispielmomentaufnahmen, um den Umgang mit Visual Explain zu lernen. Informationen zum Erstellen eigener Momentaufnahmen erhalten Sie in den folgenden Abschnitten, Sie müssen jedoch keine eigenen Momentaufnahmen erstellen, um mit dieser Lerneinheit zu arbeiten.

- Erstellen von EXPLAIN-Momentaufnahmen für dynamische SQL-Anweisungen
- Erstellen von EXPLAIN-Momentaufnahmen für statische SQL-Anweisungen

Die für die Beispielmomentaufnahmen verwendete Abfrage listet den Namen, die Abteilung sowie das Gehalt für alle Mitarbeiter auf, die keine Führungskräfte sind und deren Gehalt mehr als 90% des Gehalts der am höchsten bezahlten Führungskraft beträgt.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Die Abfrage besteht aus zwei Teilen:

1. Die Unterabfrage (in runden Klammern) produziert Datenzeilen, die 90% des Gehalts jeder Führungskraft enthalten. Da die Unterabfrage durch ALL qualifiziert ist, wird nur der größte Wert aus dieser Tabelle abgerufen.
2. Die Hauptabfrage verknüpft alle Zeilen in den Tabellen ORG und STAFF, in denen die Abteilungsnummern identisch sind, JOB nicht gleich 'Mgr' ist und das Gehalt zuzüglich Provision größer als der Wert ist, der von der Unterabfrage zurückgegeben wurde.

Die Hauptabfrage enthält die folgenden drei Vergleichselemente (Vergleiche):

1. O.DEPTNUMB = S.DEPT
2. S.JOB <> 'Mgr'
3. S.SALARY+S.COMM > ALL (SELECT ST.SALARY*.9
 FROM STAFF ST
 WHERE ST.JOB='Mgr')

Diese Vergleichselemente entsprechen jeweils Folgendem:

1. Ein Verknüpfungsvergleichselement, das die Tabellen ORG und STAFF für die Einträge verknüpft, bei denen die Abteilungsnummern identisch sind.
2. Ein lokales Vergleichselement für die Spalte JOB der Tabelle STAFF.
3. Ein lokales Vergleichselement für die Spalten SALARY und COMM der Tabelle STAFF, die das Ergebnis der Unterabfrage verwendet.

Gehen Sie wie folgt vor, um die Beispielmomentaufnahmen zu laden:

1. Wenn DB2 noch nicht gestartet wurde, setzen Sie den Befehl **db2start** ab.
2. Stellen Sie sicher, dass die EXPLAIN-Tabellen in Ihrer Datenbank vorhanden sind. Befolgen Sie hierzu die Anweisungen in *Erstellen von EXPLAIN-Tabellen*.
3. Stellen Sie eine Verbindung zu der Datenbank her, die Sie verwenden möchten. In dieser Lerneinheit stellen Sie eine Verbindung zur Beispieldatenbank SAMPLE her. Setzen Sie an der Eingabeaufforderung des DB2-Befehlszeilenprozessors den Befehl **connect to sample** ab, um eine Verbindung zur Beispieldatenbank herzustellen.

Falls die Beispieldatenbank noch nicht erstellt wurde, finden Sie die entsprechenden Informationen im Abschnitt zur Installation der Beispieldatenbank im Handbuch *Systemverwaltung*.

4. Führen Sie die DB2-Befehlsdatei VESAMPL.DDL aus, um die vordefinierten Momentaufnahmen zu importieren.

-  Diese Datei befindet sich im Verzeichnis sqllib\samples\ve.
-  Diese Datei befindet sich im Verzeichnis sqllib\samples\ve\inter.

Wechseln Sie in dieses Verzeichnis, und setzen Sie den Befehl **db2 -tf vesAMPL.ddl** ab, um die Befehlsdatei auszuführen.

- Diese Befehlsdatei muss mit derselben Benutzer-ID ausgeführt werden, die zur Erstellung der EXPLAIN-Tabellen verwendet wurde.
- Mit dieser Befehlsdatei werden nur die vordefinierten Momentaufnahmen importiert. Es werden keine Tabellen oder Daten erstellt. Die Optimierungsaktivitäten, die später beschrieben werden, (z. B. CREATE INDEX und runstats) werden für Tabellen und Daten in der Beispieldatenbank ausgeführt.

Sie können nun die Zugriffsplandiagramme anzeigen und verwenden..

Erstellen von EXPLAIN-Momentaufnahmen für dynamische SQL-Anweisungen

Anmerkung: Die Informationen zum Erstellen von EXPLAIN-Momentaufnahmen in diesem Abschnitt dienen Ihnen als Referenz. Da Ihnen Beispiel-EXPLAIN-Momentaufnahmen zur Verfügung gestellt werden, ist es nicht notwendig, diese Task durchzuführen, um mit diesem Lernprogramm zu arbeiten.

Befolgen Sie diese Schritte, um eine EXPLAIN-Momentaufnahme für eine dynamische SQL-Anweisung zu erstellen:

1. Wenn DB2 noch nicht gestartet wurde, setzen Sie den Befehl **db2start** ab.
2. Stellen Sie sicher, dass die EXPLAIN-Tabellen in Ihrer Datenbank vorhanden sind. Befolgen Sie hierzu die Anweisungen in *Erstellen von EXPLAIN-Tabellen*.
3. Stellen Sie über die Eingabeaufforderung des DB2-Befehlszeilenprozessors eine Verbindung zu der Datenbank her, die Sie verwenden möchten. Wenn Sie beispielsweise eine Verbindung zur Beispieldatenbank herstellen möchten, setzen Sie den Befehl **connect to sample** ab. Informationen zum Erstellen der Beispieldatenbank finden Sie im Abschnitt zur Installation der Beispieldatenbank im Handbuch *Systemverwaltung*.
4. Erstellen Sie eine EXPLAIN-Momentaufnahme für eine dynamische SQL-Anweisung, indem Sie einen der folgenden Befehle an der Eingabeaufforderung des DB2-Befehlszeilenprozessors verwenden:
 - Setzen Sie den Befehl **set current explain snapshot=explain** ab, um eine EXPLAIN-Momentaufnahme zu erstellen, ohne die SQL-Anweisung auszuführen.
 - Setzen Sie den Befehl **set current explain snapshot=yes** ab, um eine EXPLAIN-Momentaufnahme zu erstellen und die SQL-Anweisung auszuführen.

Mit diesem Befehl wird das EXPLAIN-Sonderregister festgelegt. Diese Festlegung gilt für alle anschließenden SQL-Anweisungen. Weitere Informationen finden Sie in den Abschnitten zu aktuellen EXPLAIN-Momentaufnahmen im Handbuch *SQL Reference*.

5. Übergeben Sie Ihre SQL-Anweisungen über die Eingabeaufforderung des DB2-Befehlszeilenprozessors.
6. Wenn Sie das Zugriffsplandiagramm für die Momentaufnahme anzeigen möchten, aktualisieren Sie das Fenster **Protokoll mit EXPLAIN bearbeiteter Anweisungen**, das Sie über die Steuerzentrale öffnen können, und klicken Sie die Momentaufnahme doppelt an.
7. Optional. Wenn Sie die Einrichtung der Momentaufnahme inaktivieren möchten, setzen Sie den Befehl **set current explain snapshot=no** ab, nachdem Sie Ihre SQL-Anweisungen übergeben haben.

Erstellen von EXPLAIN-Momentaufnahmen für statische SQL-Anweisungen

Anmerkung: Die Informationen zum Erstellen von EXPLAIN-Momentaufnahmen in diesem Abschnitt dienen Ihnen als Referenz. Da Ihnen Beispiel-EXPLAIN-Momentaufnahmen zur Verfügung gestellt werden, ist es nicht notwendig, diese Task durchzuführen, um mit diesem Lernprogramm zu arbeiten.

Befolgen Sie diese Schritte, um eine EXPLAIN-Momentaufnahme für eine statische SQL-Anweisung zu erstellen:

1. Wenn DB2 noch nicht gestartet wurde, setzen Sie den Befehl **db2start** ab.
2. Stellen Sie sicher, dass die EXPLAIN-Tabellen in Ihrer Datenbank vorhanden sind. Befolgen Sie hierzu die Anweisungen in *Erstellen von EXPLAIN-Tabellen*.
3. Stellen Sie über die Eingabeaufforderung des DB2-Befehlszeilenprozessors eine Verbindung zu der Datenbank her, die Sie verwenden möchten. Wenn Sie beispielsweise eine Verbindung zur Beispieldatenbank herstellen möchten, setzen Sie den Befehl **connect to sample** ab.
4. Erstellen Sie eine EXPLAIN-Momentaufnahme für eine statische SQL-Anweisung, indem Sie die Option EXPLSNAP verwenden, wenn Sie Ihre Anwendung binden oder vorbereiten. Setzen Sie beispielsweise den Befehl **bind ihre datei explsnap yes** ab.
5. Optional. Wenn Sie das Zugriffsplandiagramm für die Momentaufnahme anzeigen möchten, aktualisieren Sie das Fenster **Protokoll mit EXPLAIN bearbeiteter Anweisungen**, das Sie über die Steuerzentrale öffnen können, und klicken Sie die Momentaufnahme doppelt an.

Weitere Informationen zur Verwendung der Option EXPLSNAP für entsprechende APIs finden Sie in den Abschnitten zu den jeweiligen APIs im Handbuch *Application Development Guide*.

Weitere Schritte

In „Lerneinheit 2. Anzeigen und Verwenden eines Zugriffsplandiagramms“ auf Seite 7, werden Sie erfahren, wie ein Zugriffsplandiagramm angezeigt wird und wie sein Inhalt zu verstehen ist.

Lerneinheit 2. Anzeigen und Verwenden eines Zugriffsplandiagramms

In dieser Lerneinheit verwenden Sie das Fenster **Zugriffsplandiagramm**, um ein Zugriffsplandiagramm anzuzeigen und zu verwenden. Ein Zugriffsplandiagramm ist eine grafische Darstellung eines Zugriffsplans. In diesem Diagramm können Sie die folgenden Details anzeigen:

- Tabellen (und ihre zugeordneten Spalten) und Indizes
- Operatoren (wie Tabellensuche, Sortierungen und Verknüpfungen)
- Tabellenbereiche und Funktionen

Sie können ein Zugriffsplandiagramm folgendermaßen anzeigen:

- Durch Auswählen aus einer Liste von zuvor mit EXPLAIN bearbeiteten Anweisungen.
- Durch Auswählen aus einer Liste von mit EXPLAIN bearbeitbaren Anweisungen in einem Paket.
- Durch dynamisches Bearbeiten einer SQL-Anweisung mit EXPLAIN.

Da Sie mit den Zugriffsplandiagrammen für die Beispiel-Momentaufnahmen arbeiten, die Sie in Lerneinheit 1 geladen haben, treffen Sie eine Auswahl aus einer Liste von zuvor mit EXPLAIN bearbeiteten Anweisungen. Weitere Informationen zu anderen Methoden der Anzeige von Zugriffsplandiagrammen finden Sie in der Hilfe von Visual Explain.

Anzeigen eines Zugriffsplandiagramms durch Auswählen aus einer Liste von zuvor mit EXPLAIN bearbeiteten SQL-Anweisungen

Gehen Sie wie folgt vor, um ein Zugriffsplandiagramm durch Auswählen aus einer Liste von zuvor mit EXPLAIN bearbeiteten Anweisungen anzuzeigen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis die Beispieldatenbank SAMPLE angezeigt wird.
2. Klicken Sie die Datenbank mit Maustaste 2 an, und wählen Sie **Protokoll mit EXPLAIN bearbeiteter Anweisungen anzeigen** aus dem Popup-Menü aus. Das Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen wird geöffnet.
3. Sie können ein Zugriffsplandiagramm nur für eine Anweisung anzeigen, für die eine EXPLAIN-Momentaufnahme vorhanden ist. Diese Anweisungen weisen den Eintrag JA in der Spalte **EXPLAIN-Momentaufnahme** auf. Klicken Sie den Eintrag doppelt an, der als Abfragenummer 1 angegeben

ist (eventuell müssen Sie nach rechts blättern, um die Spalte **Abfragenummer** zu finden). Das Fenster **Zugriffsplandiagramm** für die Anweisung wird geöffnet.

Anmerkung: Das Diagramm wird von unten nach oben gelesen. Der erste Schritt der Abfrage wird im unteren Teil des Diagramms aufgeführt, während der letzte Schritt im oberen Teil erscheint.

Verstehen der Symbole in einem Zugriffsplandiagramm

Das Zugriffsplandiagramm zeigt die Struktur eines Zugriffsplans als Baumstruktur an. Die *Knoten* der Baumstruktur stellen Folgendes dar:

- Tabellen, als Rechtecke dargestellt
- Indizes, als Rauten dargestellt
- Operatoren, als Achtecke dargestellt. TQUEUE-Operatoren, als Parallelogramme dargestellt
- Tabellenfunktionen, als Sechsecke dargestellt

Bei Operatoren ist die Zahl in Klammern rechts neben dem Operortyp eine eindeutige Kennung für jeden Knoten. Die Zahl unter dem Operortyp stellt den kumulativen Aufwand dar.

Verwenden der Zoomleiste zum Vergrößern von Teilen eines Diagramms

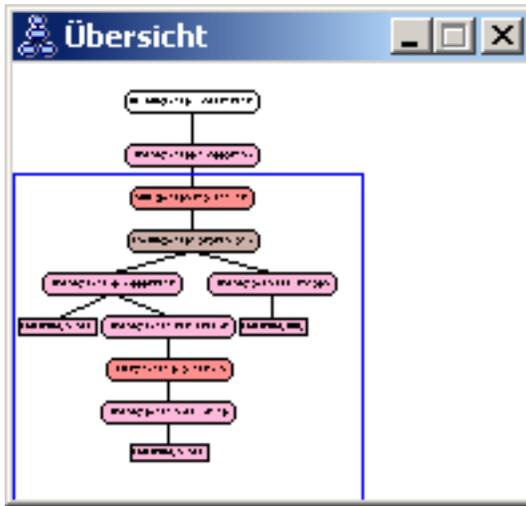
Wenn Sie ein Zugriffsplandiagramm anzeigen, wird das gesamte Diagramm angezeigt, und Sie können eventuell die Details zu den einzelnen Knoten nicht sehen.

Verwenden Sie im Fenster **Zugriffsplandiagramm** die Zoomleiste, um Teile eines Diagramms zu vergrößern:

1. Positionieren Sie den Mauszeiger über dem kleinen Schiebefeld in der Zoomleiste links neben dem Diagramm.
2. Klicken Sie die Leiste an, und ziehen Sie sie, bis das Diagramm die gewünschte Vergrößerung erreicht hat.

Wenn Sie verschiedene Teile des Diagramms anzeigen möchten, verwenden Sie die Schiebeleiste.

Wenn Sie ein großes und kompliziertes Zugriffsplandiagramm anzeigen möchten, verwenden Sie das Fenster **Diagrammübersicht**. Sie können dieses Fenster verwenden, um zu sehen, welchen Teil des Diagramms Sie gerade anzeigen, und um das Diagramm zu vergrößern oder darin zu blättern. Der Abschnitt im Zoomrahmen wird im Zugriffsplan angezeigt.



Um verschiedene Teile des Diagramms anzuzeigen, positionieren Sie den Mauszeiger über dem hervorgehobenen Bereich im Fenster **Übersicht** und bewegen die Maus mit gedrückter Maustaste 1, bis der gewünschte Teil des Zugriffsplandiagramms zu sehen ist.

Anzeigen weiterer Details zu den Objekten in einem Diagramm

Sie können auf weitere Informationen zu den Objekten in einem Zugriffsplandiagramm zugreifen. Sie können Folgendes anzeigen:

- Systemkatalogstatistiken für Objekte wie:
 - Tabellen, Indizes oder Tabellenfunktionen
 - Informationen zu Operatoren, wie deren Aufwand, Merkmale und Eingabeargumente
 - Integrierte Funktionen oder benutzerdefinierte Funktionen
 - Tabellenbereiche
 - Spalten, auf die in einer SQL-Anweisung verwiesen wird
- Informationen zu Konfigurationsparametern und Bindeoptionen (Optimierungsparameter)

Anzeigen von Statistikdaten für Tabellen, Indizes und Tabellenfunktionen

Wenn Sie Katalogstatistiken für eine Tabelle (Rechteck), einen Index (Raute) oder eine Tabellenfunktion (Sechseck) in einem Diagramm anzeigen möchten, klicken Sie den entsprechenden Knoten doppelt an. Für die ausgewählten Objekte wird ein Statistikfenster geöffnet, in dem Informationen zu den Statistiken angezeigt werden, die gültig waren, als die Momentaufnahme erstellt wurde, sowie zu den Statistiken, die derzeit in den Systemkatalogtabellen vorhanden sind.

Wenn Sie Katalogstatistiken für *mehrere* Tabellen, Indizes oder Tabellenfunktionen in einem Diagramm anzeigen möchten, wählen Sie jedes einzelne Objekt aus, indem Sie es anklicken (es wird hervorgehoben), und wählen Sie anschließend **Knoten→Statistikdaten anzeigen** aus. Für jedes ausgewählte Objekt wird ein Statistikfenster geöffnet. (Die Fenster können übereinander liegen und müssen eventuell verschoben werden, damit auf alle Fenster zugegriffen werden kann).

Wenn der Eintrag für **STATS_TIME** in der Spalte **Mit EXPLAIN bearbeitet** den Wert **Statistikdaten nicht aktualisiert** hat, waren keine Statistikdaten vorhanden, als das Optimierungsprogramm den Zugriffsplan erstellt hat. Daher wurden vom Optimierungsprogramm die Standardwerte verwendet, wenn zur Erstellung eines Zugriffsplans bestimmte Statistikdaten erforderlich waren. Wenn vom Optimierungsprogramm Standardstatistikdaten verwendet wurden, wird für sie (**Standardwert**) in der Spalte **Mit EXPLAIN bearbeitet** angegeben.

Anzeigen von Details zu Operatoren in einem Diagramm

Wenn Sie Katalogstatistiken für einen einzelnen Operator (Achteck) anzeigen möchten, klicken Sie seinen Knoten doppelt an. Für den ausgewählten Operator wird das Fenster **Operatordetails** geöffnet, in dem folgende Informationen angezeigt werden:

- Der geschätzte kumulative Aufwand (E/A, CPU-Anweisungen und Gesamtaufwand)
- Die bisherige Kardinalität (d. h. die geschätzte Anzahl der durchsuchten Zeilen)
- Die Tabellen, auf die bisher im Zugriffsplan zugegriffen wurde und die verknüpft wurden
- Die Spalten der Tabellen, auf die bisher zugegriffen wurde
- Die Vergleichselemente, die bisher angewendet wurden, einschließlich ihrer geschätzten Selektivität
- Die Eingabeargumente für jeden Operator

Wenn Sie Details zu *mehreren* Operatoren anzeigen möchten, wählen Sie jeden einzelnen Operator aus, indem Sie ihn anklicken (er wird hervorgehoben), und wählen Sie anschließend **Knoten→Details anzeigen** aus. Für jedes ausgewählte Objekt wird ein Statistikfenster geöffnet. (Die Fenster können übereinander liegen und müssen eventuell verschoben werden, damit auf alle Fenster zugegriffen werden kann).

Anzeigen von Statistikdaten für Funktionen

Wenn Sie Katalogstatistiken für integrierte Funktionen und benutzerdefinierte Funktionen anzeigen möchten, wählen Sie **Anweisung→Statistikdaten anzeigen→Funktionen** aus. Wählen Sie mindestens einen Eintrag aus der Liste aus, die im Fenster **Funktionen** angezeigt wird, und klicken Sie **OK** an. Für jede ausgewählte Funktion wird das Fenster **Funktionsstatistiken** geöffnet.

Anzeigen von Statistikdaten für Tabellenbereiche

Wenn Sie Katalogstatistiken für Tabellenbereiche anzeigen möchten, wählen Sie **Anweisung->Statistikdaten anzeigen->Tabellenbereiche** aus. Wählen Sie mindestens einen Eintrag aus der Liste aus, die im Fenster **Tabellenbereiche** angezeigt wird, und klicken Sie **OK** an. Für jeden ausgewählten Tabellenbereich wird das Fenster **Tabellenbereichsstatistik** geöffnet.

Anzeigen von Statistikdaten für Spalten in einer SQL-Anweisung

Gehen Sie wie folgt vor, um Statistikdaten für die Spalten anzuzeigen, auf die in einer SQL-Anweisung verwiesen wird:

1. Klicken Sie eine Tabelle im Zugriffsplandiagramm doppelt an. Das Fenster **Tabellenstatistik** wird geöffnet.
2. Klicken Sie den Druckknopf **Zugehörige Spalten** an. Das Fenster **Zugehörige Spalten** wird geöffnet, in dem die Spalten der Tabelle aufgelistet sind.
3. Wählen Sie mindestens eine Spalte aus der Liste aus, und klicken Sie **OK** an. Für jede ausgewählte Spalte wird das Fenster **Zugehörige Spaltenstatistiken** geöffnet.

Anzeigen von Informationen zu Konfigurationsparametern und Bindeoptionen

Wenn Sie Informationen zu Konfigurationsparametern und Bindeoptionen (Optimierungsparameter) anzeigen möchten, wählen Sie **Anweisung->Optimierungsparameter anzeigen** im Fenster **Zugriffsplandiagramm** aus. Das Fenster **Optimierungsparameter** wird geöffnet, in dem Informationen zu den Parameterwerten angezeigt werden, die zum Zeitpunkt der Erstellung der Momentaufnahme vorlagen, sowie zu den aktuellen Werten.

Ändern der Darstellung eines Diagramms

Gehen Sie wie folgt vor, um verschiedene Kenndaten einer Diagrammdarstellung zu ändern:

1. Wählen Sie **Sicht->Einstellungen** im Fenster **Zugriffsplandiagramm** aus. Das Notizbuch **Einstellungen des Zugriffsplandiagramms** wird geöffnet.
2. Wenn Sie die Hintergrundfarbe ändern möchten, wählen Sie die Indexzunge **Diagramm** aus.
3. Wenn Sie die Farbe verschiedener Operatoren ändern möchten, verwenden Sie die Indexzungen **Basis**, **Erweitern**, **Aktualisieren** und **Verschiedenes**.
4. Wenn Sie die Farbe der Tabellen-, Index- oder Tabellenfunktionsknoten ändern möchten, wählen Sie die Indexzunge **Operand** aus.
5. Wählen Sie die Indexzunge **Operator** aus, um anzugeben, welche Informationen in den Operatorknoten angezeigt werden (Typ des Aufwands oder Kardinalität, d. h. die geschätzte Zahl der bisher zurückgegebenen Zeilen).
6. Wählen Sie die Indexzunge **Operand** aus, um anzugeben, ob die Schemanamen oder Benutzer-IDs in den Tabellenknoten angezeigt werden.

7. Wählen Sie die Indexzunge **Knoten** aus, um anzugeben, ob die Knoten zweidimensional oder dreidimensional angezeigt werden sollen.
8. Klicken Sie **Anwenden** an, um das Diagramm mit den ausgewählten Optionen zu aktualisieren und die Einstellungen zu speichern.

Weitere Schritte

Wenn Sie in einer Umgebung mit Datenbanken mit einer Partition arbeiten, gehen Sie zu „Lerneinheit 3. Verbessern eines Zugriffsplans in einer Umgebung mit Datenbanken mit einer Partition“ auf Seite 13, wo Sie erfahren, wie verschiedene Aktivitäten zur Optimierung einen Zugriffsplan ändern und verbessern können.

Wenn Sie in einer Umgebung mit partitionierten Datenbanken arbeiten, gehen Sie zu „Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten“ auf Seite 14, wo Sie erfahren, wie verschiedene Aktivitäten zur Optimierung einen Zugriffsplan ändern und verbessern können.

Lerneinheit 3. Verbessern eines Zugriffsplans in einer Umgebung mit Datenbanken mit einer Partition

In dieser Lerneinheit erfahren Sie, wie sich der Zugriffsplan und damit verbundene Fenster für die Standardabfrage ändern, wenn Sie verschiedene Optimierungsaktivitäten ausführen. Mit Hilfe von einigen illustrierten Beispielen erfahren Sie, wie der geschätzte Gesamtaufwand für einen Zugriffsplan auch bei einfachen Abfragen verbessert werden kann, indem der Befehl **runstats** verwendet wird und entsprechende Indizes hinzugefügt werden.

Mit zunehmender Erfahrung im Umgang mit Visual Explain werden Sie noch weitere Möglichkeiten zur Optimierung von Abfragen entdecken.

Arbeiten mit Zugriffsplandiagrammen

Anhand der vier Beispiel-EXPLAIN-Momentaufnahmen werden Sie erfahren, dass die Optimierung eine wichtige Rolle bei der Datenbankleistung spielt.

Die den EXPLAIN-Momentaufnahmen zugeordneten Abfragen sind mit 1–4 numeriert. Jede Abfrage verwendet dieselbe SQL-Anweisung (wie in Lerneinheit 1 beschrieben):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Aber jede Iteration der Abfrage verwendet mehr Optimierungstechniken als die vorherige Ausführung. Bei Abfrage 1 gab es beispielsweise gar keine Leistungsverbesserung, während es bei Abfrage 4 die meisten Leistungsverbesserungen gab. Die Unterschiede bei den Abfragen werden im Folgenden beschrieben:

Abfrage 1

Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten

Abfrage 2

Erfassen aktueller Statistikdaten für die Tabellen und Indizes in einer Abfrage

Abfrage 3

Erstellen von Indizes für Spalten, die zur Verknüpfung von Tabellen in einer Abfrage verwendet werden

Abfrage 4

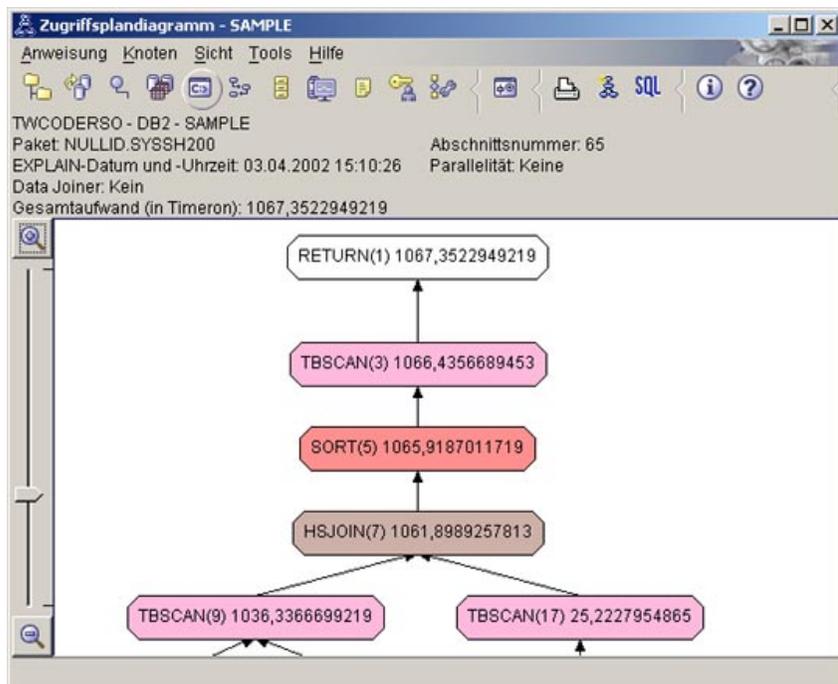
Erstellen zusätzlicher Indizes für Tabellenspalten

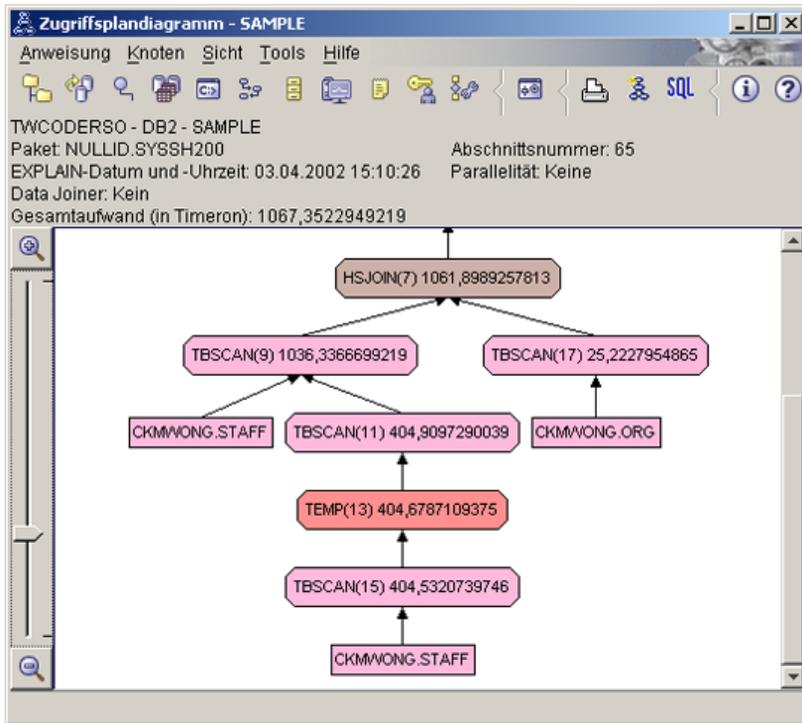
Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten

In diesem Beispiel wurde der Zugriffsplan für die SQL-Abfrage ohne Indizes und ohne Statistikdaten erstellt.

Gehen Sie wie folgt vor, um das Zugriffsplandiagramm für diese Abfrage (Abfrage 1) anzuzeigen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis die Beispieldatenbank SAMPLE angezeigt wird.
2. Klicken Sie die Datenbank mit Maustaste 2 an, und wählen Sie **Protokoll mit EXPLAIN bearbeiteter Anweisungen anzeigen** aus. Das Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen wird geöffnet.
3. Klicken Sie den Eintrag doppelt an, der als Abfragenummer 1 angegeben ist (eventuell müssen Sie nach rechts blättern, um die Spalte **Abfragenummer** zu finden). Das Fenster **Zugriffsplandiagramm** wird für die Anweisung geöffnet.





Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

1. Sind für jede Tabelle in der Abfrage aktuelle Statistikdaten vorhanden?

Wenn Sie überprüfen möchten, ob aktuelle Statistikdaten für jede Tabelle in der Abfrage vorhanden sind, klicken Sie jeden Tabellenknoten im Zugriffsplandiagramm doppelt an. Im Fenster **Tabellenstatistik**, das nun geöffnet wird, enthält die Zeile **STATS_TIME** in der Spalte **Mit EXPLAIN bearbeitet** die Worte "Statistikdaten nicht aktualisiert", wenn zu dem Zeitpunkt, als die Momentaufnahme erstellt wurde, keine Statistikdaten erfasst waren.

Wenn keine aktuellen Statistikdaten vorhanden sind, verwendet das Optimierungsprogramm Standardstatistikdaten, die mit den tatsächlichen Statistikdaten eventuell nicht identisch sind. Standardstatistikdaten werden mit dem Wort "Standardwert" in der Spalte **Mit EXPLAIN bearbeitet** im Fenster **Tabellenstatistik** angegeben.

Laut den Informationen im Fenster **Tabellenstatistik** für die Tabelle **ORG** hat das Optimierungsprogramm Standardstatistikdaten verwendet (wie neben den mit EXPLAIN bearbeiteten Werten angegeben).

Es wurden Standardstatistikdaten verwendet, da tatsächliche Statistikdaten nicht zur Verfügung standen, als die Momentaufnahme erstellt wurde (wie in der Zeile **STATS_TIME** angegeben).

Statistik	Mit EXPLAIN bearbeitet	Aktuell
CREATE_TIME	03.04.2002 15:05:03	
STATS_TIME	Statistikdaten nicht aktualisiert	Statistikdaten nicht aktualisiert
CARD	55(Standardwert)	
NPAGES	1(Standardwert)	
FPAGES	1(Standardwert)	
COLCOUNT	5(Standardwert)	
OVERFLOW	0(Standardwert)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
FLÜCHTIG	Kein(Standardwert)	Kein

2. Verwendet dieser Zugriffsplan die effektivsten Methoden für den Datenzugriff?

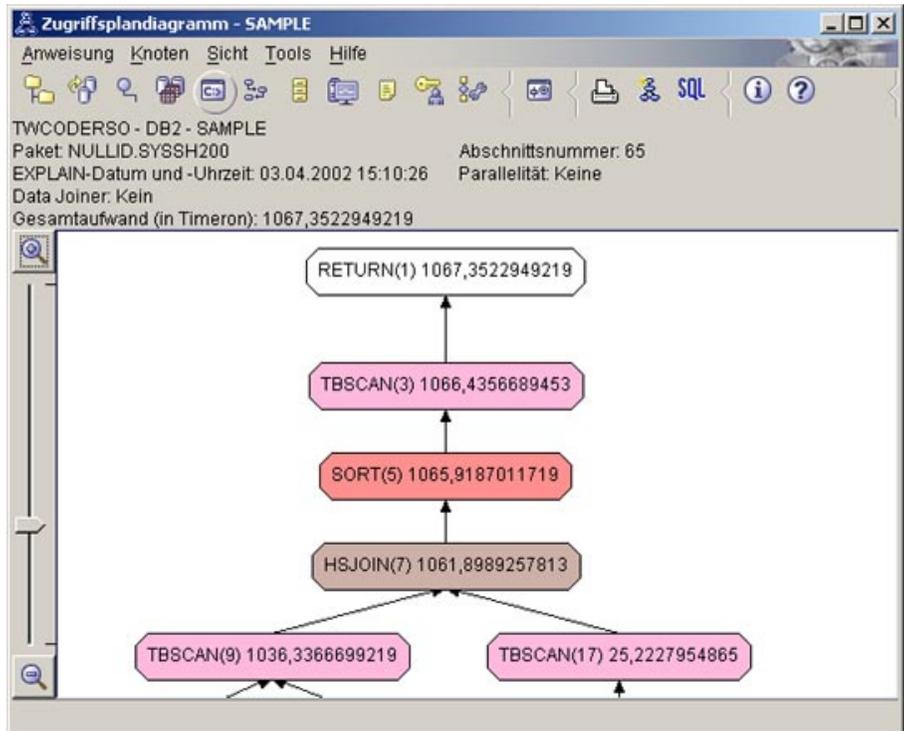
Dieser Zugriffsplan enthält Tabellensuchen anstelle von Indexsuchen. Eine Tabellensuche wird als Achteck angezeigt und mit TBSCAN angegeben. Falls eine Indexsuche verwendet worden wäre, würde sie als Raute angezeigt und mit IXSCAN angegeben. Die Verwendung eines Indexes, der für eine Tabelle erstellt wurde, ist nicht so aufwändig wie eine Tabellensuche, wenn kleine Datenmengen extrahiert werden.

3. Wie effektiv ist dieser Zugriffsplan?

Sie können die Effektivität eines Zugriffsplans nur feststellen, wenn er auf tatsächlichen Statistikdaten basiert. Da das Optimierungsprogramm im Zugriffsplan Standardstatistikdaten verwendet hat, können Sie die Effektivität des Plans nicht feststellen.

Sie sollten sich grundsätzlich den gesamten geschätzten Aufwand für den Zugriffsplan notieren, um ihn später mit überarbeiteten Zugriffsplänen vergleichen zu können. Der in jedem Knoten aufgeführte Aufwand ist kumulativ, vom ersten Schritt Ihrer Abfrage bis einschließlich des Knotens.

Im Fenster **Zugriffsplandiagramm** beträgt der Gesamtaufwand etwa 1.067 Timeron, wie unter **RETURN(1)** im oberen Bereich des Diagramms angezeigt. Der gesamte geschätzte Aufwand wird ebenfalls im oberen Bereich des Fensters angezeigt.



4. Weitere Schritte

Abfrage 2 bezieht sich auf einen Zugriffspl für die Standardabfrage, nachdem der Befehl **runstats** ausgeführt wurde. Wenn Sie den Befehl **runstats** verwenden, werden dem Optimierungsprogramm die aktuellen Statistikdaten für alle Tabellen zur Verfügung gestellt, auf die die Abfrage zugreift.

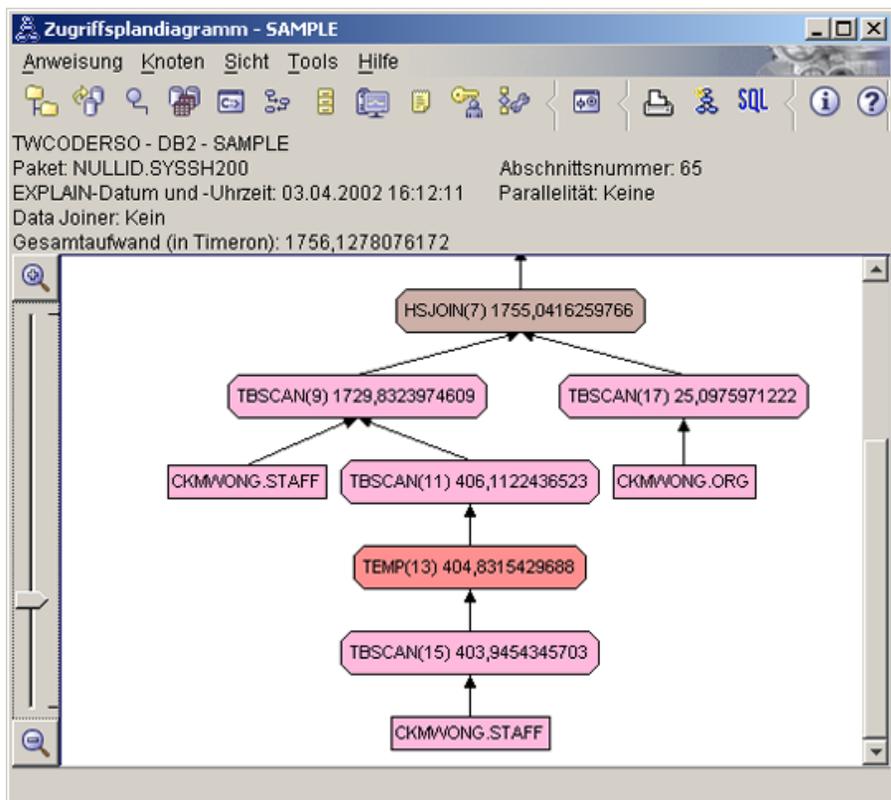
Erfassen aktueller Statistikdaten für die Tabellen und Indizes mit Hilfe des Befehls 'runstats'

Dieses Beispiel baut auf dem Zugriffspl auf, der in Abfrage 1 beschrieben wurde, indem aktuelle Statistikdaten mit dem Befehl **runstats** erfasst werden.

Es wird dringend empfohlen, die aktuellen Statistikdaten für Tabellen und Indizes mit dem Befehl **runstats** zu erfassen, insbesondere, wenn seit der letzten Ausführung des Befehls **runstats** umfangreiche Aktualisierungen vorgenommen oder neue Indizes erstellt wurden. Dadurch werden dem Optimierungsprogramm die genauesten Informationen zur Verfügung gestellt, anhand derer es den besten Zugriffspl ermitteln kann. Wenn aktuelle Statistikdaten nicht verfügbar sind, kann das Optimierungsprogramm einen nicht effizienten Zugriffspl auswählen, der auf ungenauen Standardstatistikdaten basiert.

Stellen Sie sicher, dass **runstats** *nach* der Aktualisierung der Tabelle ausgeführt wird. Andernfalls hält das Optimierungsprogramm die Tabelle für leer. Dies ist der Fall, wenn als Kardinalität im Fenster **Operatordetails** Null angegeben ist. In diesem Fall beenden Sie Ihre Tabellenaktualisierungen, führen den Befehl **runstats** erneut aus und erstellen erneut die EXPLAIN-Momentaufnahmen für betroffene Tabellen.

Wenn Sie das Zugriffsplandiagramm für diese Abfrage (Abfrage 2) anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 2 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.



Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

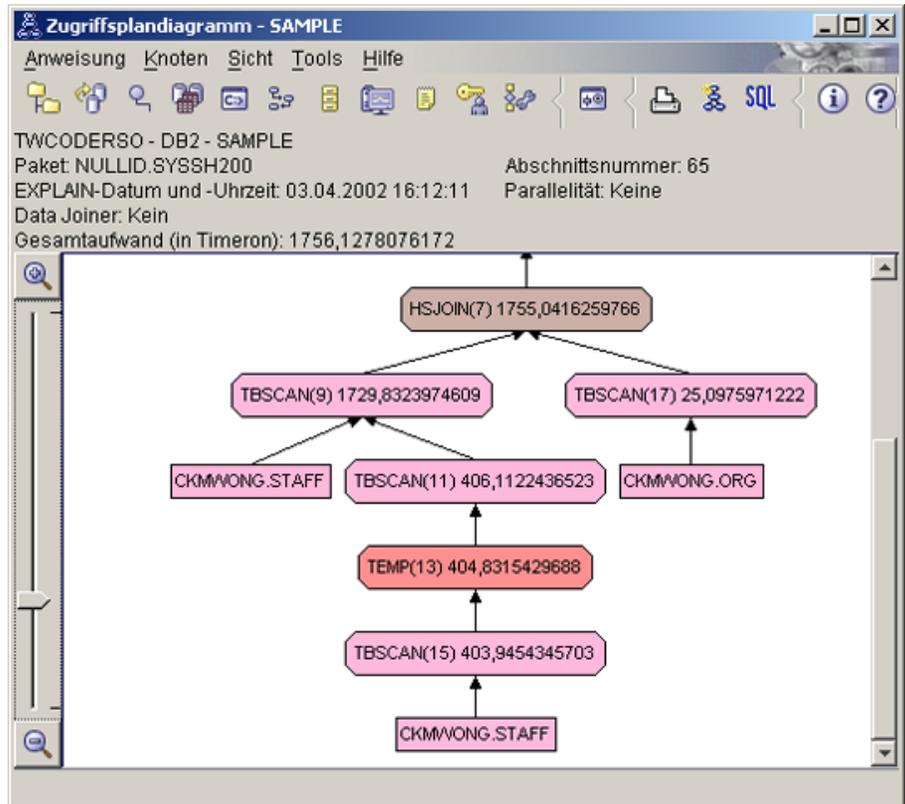
1. Sind für jede Tabelle in der Abfrage aktuelle Statistikdaten vorhanden?

Das Fenster **Tabellenstatistik** für die Tabelle ORG zeigt an, dass das Optimierungsprogramm tatsächliche Statistikdaten verwendet hat (der Wert **STATS_TIME** stellt die Zeit dar, zu der die Statistikdaten tatsächlich erfasst wurden). Die Genauigkeit der Statistikdaten hängt davon ab, ob der Inhalt der Tabellen seit der Ausführung des Befehls **runstats** erheblich geändert wurde.

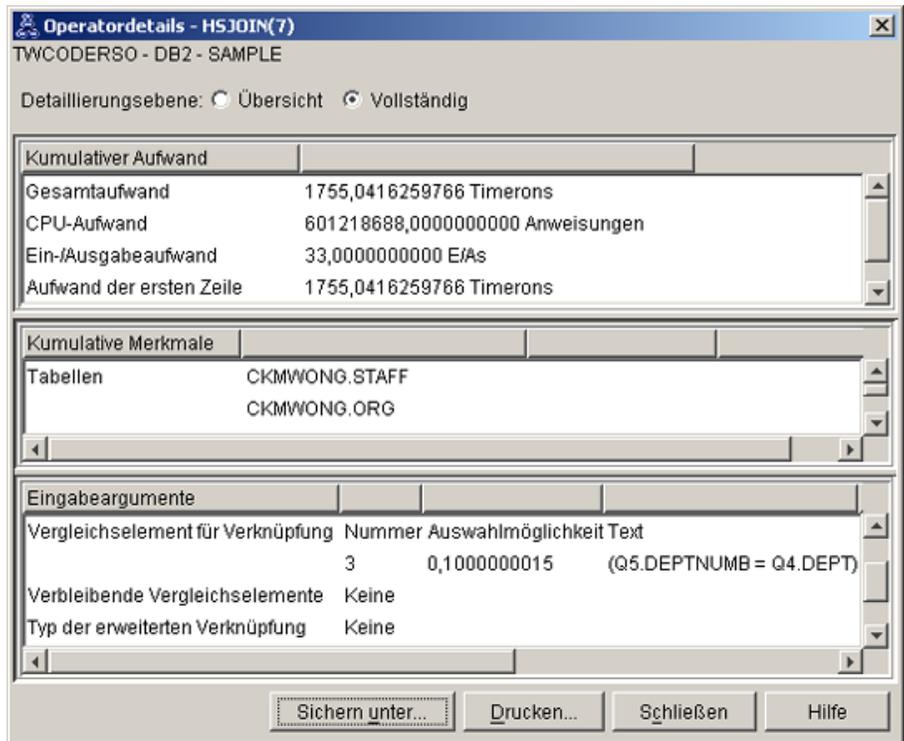
Statistik	Mit EXPLAIN bearbeitet	Aktuell
CREATE_TIME	03.04.2002 15:05:03	
STATS_TIME	03.04.2002 16:12:05	Statistikdaten nicht aktualisiert
CARD	8	
NPAGES	1	
FPAGES	1	
COLCOUNT	5	
OVERFLOW	0	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
FLÜCHTIG	Kein	Kein

2. Verwendet dieser Zugriffsplan die effektivsten Methoden für den Datenzugriff?

Wie bei Abfrage 1 verwendet der Zugriffsplan in Abfrage 2 Tabellensuchen (TBSCAN) anstelle von Indexsuchen (IXSCAN). Obwohl aktuelle Statistikdaten vorhanden sind, wurde keine Indexsuche vorgenommen, da es keine Indizes für die Spalten gibt, die von der Abfrage verwendet wurden. Die Abfrage kann verbessert werden, indem dem Optimierungsprogramm Indizes für Spalten zur Verfügung gestellt werden, die zur Verknüpfung von Tabellen verwendet werden (d. h. für Spalten, die in Verknüpfungselementen verwendet werden). In diesem Beispiel ist dies die erste Mischsuchverknüpfung: HSJOIN(7).



Sehen Sie sich im Fenster **Operatordetails** für den Operator HSJOIN(7) den Abschnitt **Vergleichselement für Verknüpfung** unter **Eingabeargumente** an. Die in dieser Verknüpfung verwendeten Spalten werden in der Spalte **Text** aufgelistet. In diesem Beispiel lauten diese Spalten DEPTNUMB und DEPT.



3. Wie effektiv ist dieser Zugriffsplan?

Zugriffspläne, die auf aktuellen Statistikdaten basieren, produzieren immer einen realistischen geschätzten Aufwand (gemessen in Timeron). Da der geschätzte Aufwand in Abfrage 1 auf Standardstatistikdaten basierte, können wir den Aufwand der beiden Zugriffsplandiagramme nicht vergleichen, um festzustellen, welcher effektiver ist. Es spielt keine Rolle, ob der Aufwand höher oder niedriger ist. Sie müssen den Aufwand von Zugriffsplänen vergleichen, die auf tatsächlichen Statistikdaten basieren, um ein gültiges Maß für die Effektivität zu erhalten.

4. Weitere Schritte

Abfrage 3 bezieht sich darauf, welche Auswirkungen das Hinzufügen von Indizes für die Spalten DEPTNUMB und DEPT hat. Wenn Sie Indizes für die Spalten hinzufügen, die in Verknüpfungsvergleichselementen verwendet werden, kann dies die Leistung erhöhen.

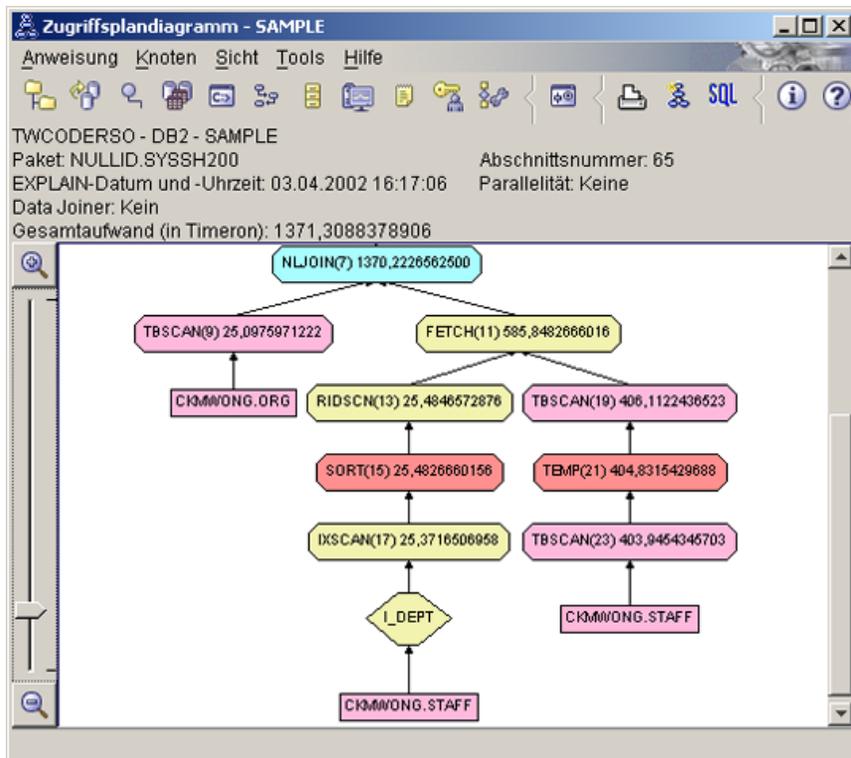
Erstellen von Indizes für Spalten, die verwendet werden, um Tabellen in einer Abfrage zu verknüpfen

Dieses Beispiel baut auf dem Zugriffsplan auf, der in Abfrage 2 beschrieben wurde, indem Indizes für die Spalte DEPT in der Tabelle STAFF und für die Spalte DEPTNUMB in der Tabelle ORG erstellt werden.

Anmerkung: In Version 8 können empfohlene Indizes mit Hilfe des Assistenten für die Auslastungsleistung erstellt werden.

Wenn Sie das Zugriffsplandiagramm für diese Abfrage (Abfrage 3) anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 3 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.

Anmerkung: Obwohl ein Index für DEPTNUM erstellt wurde, wurde er vom Optimierungsprogramm nicht verwendet.

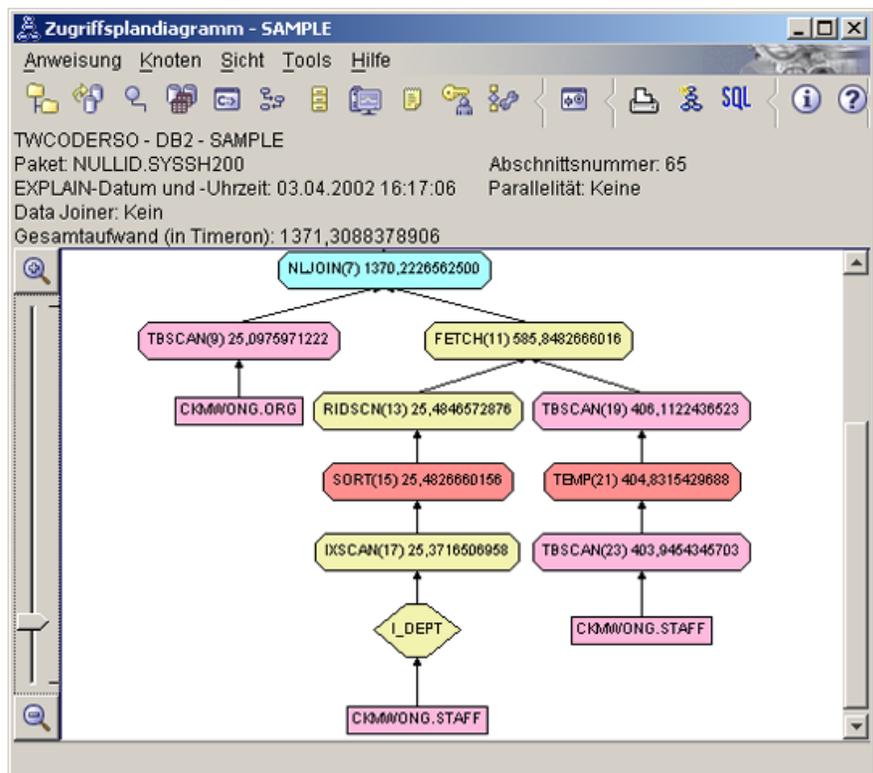


Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

1. Was hat sich im Zugriffsplan mit Indizes geändert?

Eine verschachtelte Schleifenverknüpfung, NLJOIN(7), hat die Mischsuchverknüpfung HSJOIN(7) ersetzt, die in Abfrage 2 verwendet wurde. Die Verwendung der verschachtelten Schleifenverknüpfung führte zu einem niedrigeren geschätzten Aufwand als eine Mischsuchverknüpfung, da dieser Verknüpfungstyp keine Sortiertabellen oder temporären Tabellen erfordert.

Es wurde ein neuer Knoten, I_DEPT, in Form einer Raute über der Tabelle STAFF hinzugefügt. Dieser Knoten stellt den Index dar, der für DEPT erstellt wurde. Er zeigt an, dass das Optimierungsprogramm eine Indexsuche anstelle einer Tabellensuche verwendet hat, um festzustellen, welche Zeilen abgerufen werden müssen.



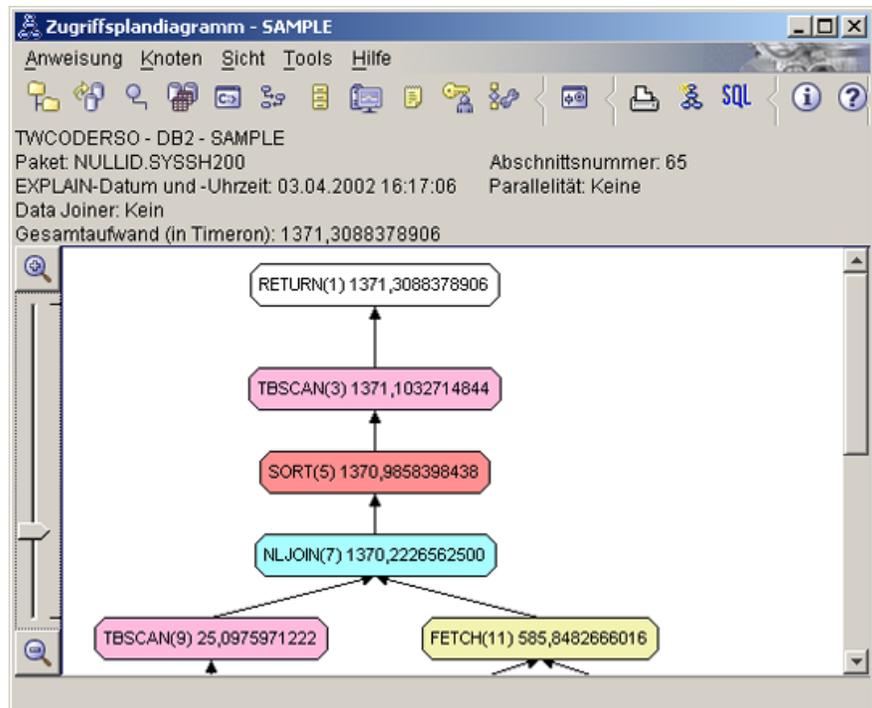
Beachten Sie in diesem Teil des Zugriffsplandiagramms, dass ein neuer Index (I_DEPT) für die Spalte DEPT erstellt wurde und IXSCAN(17) für den Zugriff auf die Tabelle STAFF verwendet wurde. In Abfrage 2 wurde eine Tabellensuche verwendet, um auf die Tabelle STAFF zuzugreifen.

2. Verwendet dieser Zugriffssplan die effektivsten Methoden für den Datenzugriff?

Aufgrund der hinzugefügten Indizes wurde der IXSCAN-Knoten IXSCAN(17) für den Zugriff auf die Tabelle STAFF verwendet. Da es in Abfrage 2 keinen Index gab, wurde dort eine Tabellensuche verwendet.

Der FETCH-Knoten FETCH(11) zeigt an, dass das Optimierungsprogramm neben der Verwendung der Indexsuche zum Abruf der Spalte DEPT zusätzliche Spalten aus der Tabelle STAFF abgerufen hat, wobei der Index als Zeiger verwendet wurde. In diesem Fall wird die Kombination von Indexsuche und Abruf als weniger aufwändig berechnet als die vollständige Tabellensuche, die in den vorherigen Zugriffsplänen verwendet wurde.

Anmerkung: Der Knoten für die Tabelle STAFF wird zweimal angezeigt, um seine Beziehung zum Index für DEPT und zur FETCH-Operation anzuzeigen.



Der Zugriffsplan für diese Abfrage zeigt an, welche Auswirkungen das Erstellen von Indizes für Spalten hat, die für Verknüpfungsvergleichselemente relevant sind. Indizes können auch die Anwendung von lokalen Vergleichselementen beschleunigen. Sehen wir uns die lokalen Vergleichselemente für jede Tabelle in dieser Abfrage an, um festzustellen, wie das Hinzufügen von Indizes zu Spalten, auf die in lokalen Vergleichselementen verwiesen wird, den Zugriffsplan beeinflusst.

Sehen Sie sich im Fenster **Operatordetails** für den Operator FETCH(11) die Spalten unter **Kumulative Merkmale** an. Die Spalte, die im Vergleichselement für diesen Abruf verwendet wird, lautet JOB, wie im Abschnitt **Vergleichselemente** angezeigt.

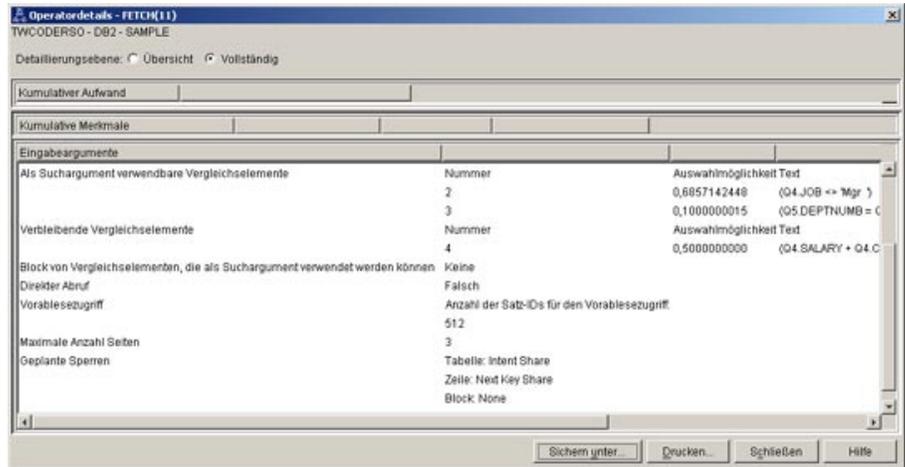
Anmerkung: Die Selektivität dieses Vergleichselements lautet 0,69. Dies bedeutet, dass mit diesem Vergleichselement 69% der Zeilen für eine weitere Verarbeitung ausgewählt werden.

The screenshot shows a window titled 'Operatordetails - FETCH(11)' with a sub-header 'TWCODERSO - DB2 - SAMPLE'. It has two radio buttons for 'Detailierungsebene': 'Übersicht' (selected) and 'Vollständig'. The window is divided into two main sections: 'Kumulativer Aufwand' and 'Kumulative Merkmale'.

Kumulativer Aufwand	
Gesamtaufwand	585,8482666016 Timerons
CPU-Aufwand	63529024,0000000000 Anweisungen
Ein-/Ausgabeaufwand	32,9705467224 E/As
Aufwand der ersten Zeile	458,8978271484 Timerons

Kumulative Merkmale			
Tabellen	CKMWONG.STAFF		
Spalten	CKMWONG.STAFF.NAME CKMWONG.STAFF.ID CKMWONG.STAFF.COMM CKMWONG.STAFF.SALARY		
Sortierspalten	Keine		
Vergleichselemente	Nummer	Auswahlmöglichkeit	Text
	2	0,6857142448	(Q4.JOB <=> 'Mgr')
	3	0,1000000015	(Q5.DEPTNUMB = Q4.DEPT
	4	0,5000000000	(Q4.SALARY + Q4.COMM) >
Kardinalität	37,1999969482		
Summe der verwendeten Pufferpoolseiten	23,3865756989		

At the bottom of the window, there are four buttons: 'Sichern unter...', 'Drucken...', 'Schließen', and 'Hilfe'.



Das Fenster **Operatordetails** für den Operator FETCH(11) zeigt die Spalten an, die in dieser Operation verwendet werden. Sie können sehen, dass DEPTNAME in der ersten Zeile neben **Abgerufene Spalten** unter **Eingabeargumente** aufgelistet ist.

3. Wie effektiv ist dieser Zugriffsplan?

Dieser Zugriffsplan ist nicht so aufwändig wie der Zugriffsplan aus dem vorherigen Beispiel. Der kumulative Aufwand wurde von etwa 1.756 Timeron in Abfrage 2 auf etwa 960 Timeron in Abfrage 3 reduziert.

Der Zugriffsplan für Abfrage 3 zeigt jedoch eine Indexsuche IXSCAN(17) und einen Abruf FETCH(11) für die Tabelle STAFF an. Obwohl eine Indexsuche in Kombination mit einer FETCH-Operation weniger aufwändig ist als eine vollständige Tabellensuche, bedeutet diese Kombination, dass für jede abgerufene Zeile auf die Tabelle und auf den Index jeweils einmal zugegriffen wird. Wir möchten versuchen, diesen doppelten Zugriff auf die Tabelle STAFF zu reduzieren.

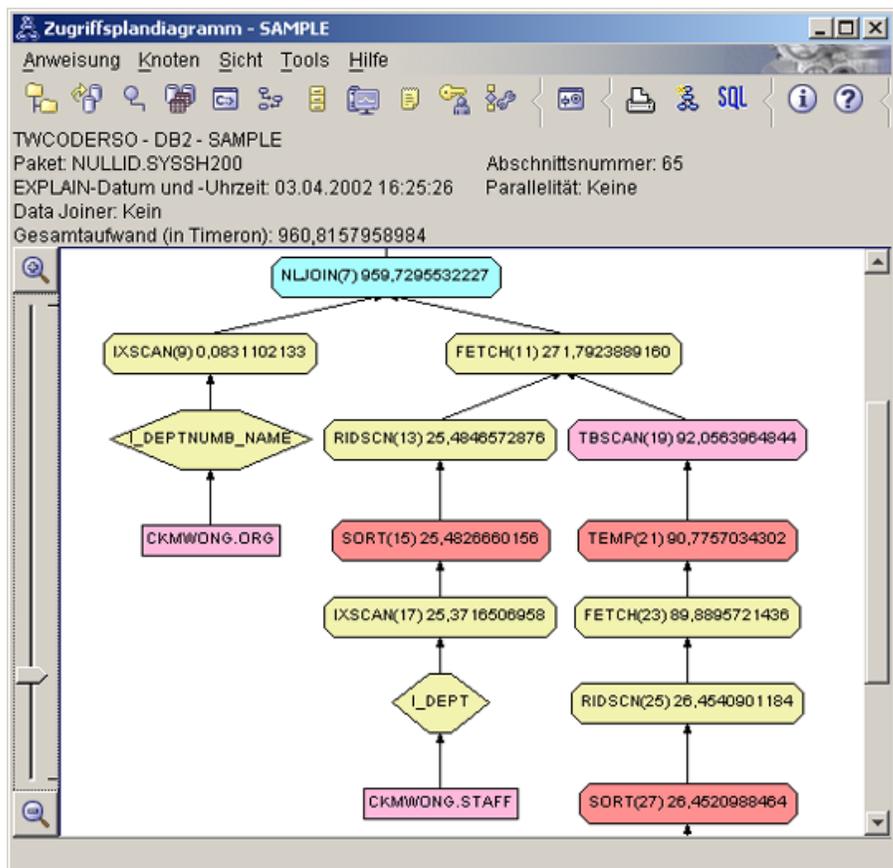
4. Weitere Schritte

Abfrage 4 reduziert die FETCH-Operation und die Indexsuche auf eine einzige Indexsuche ohne FETCH-Operation. Durch das Erstellen zusätzlicher Indizes kann der geschätzte Aufwand für den Zugriffsplan reduziert werden.

Erstellen zusätzlicher Indizes für Tabellenspalten

Dieses Beispiel baut auf dem Zugriffsplan auf, der in Abfrage 3 beschrieben wurde, indem ein Index für die Spalte JOB in der Tabelle STAFF erstellt wird und indem DEPTNAME zum vorhandenen Index in der Tabelle ORG hinzugefügt wird. (Das Hinzufügen eines separaten Indexes könnte zu einem zusätzlichen Zugriff führen.)

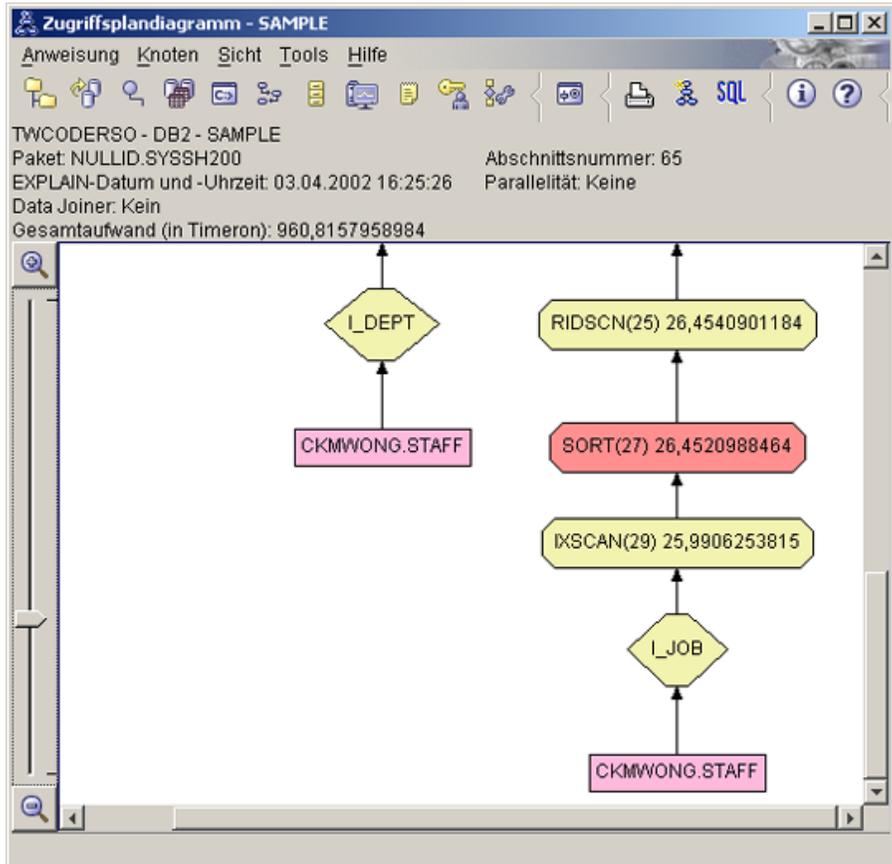
Wenn Sie das Zugriffsplandiagramm für diese Abfrage (Abfrage 4) anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 4 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.



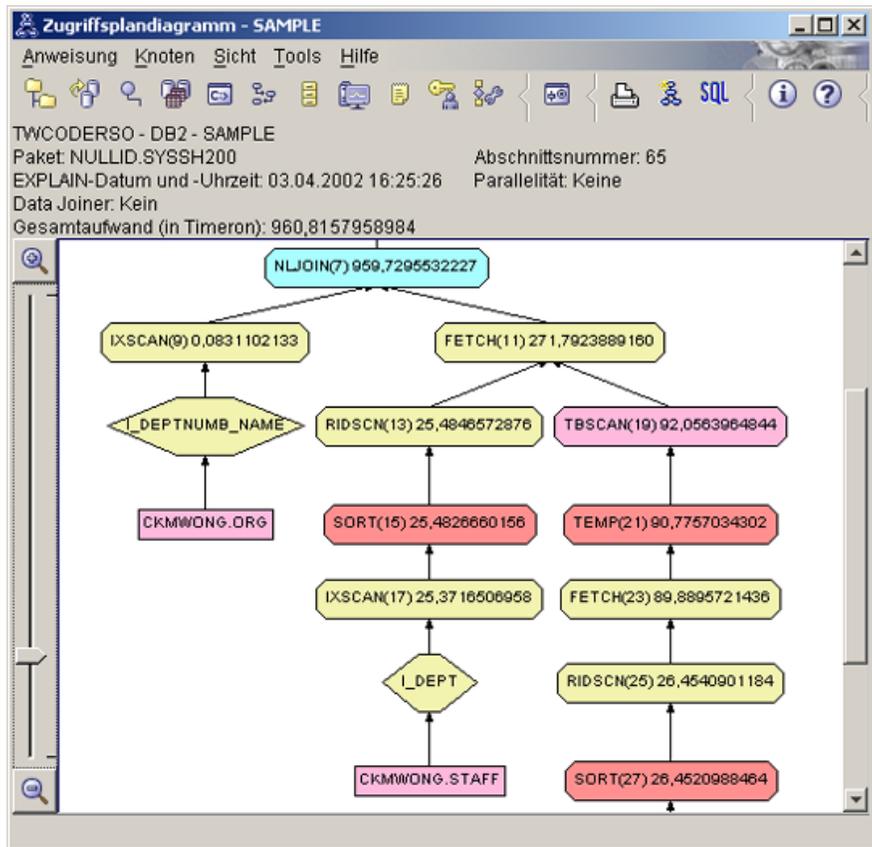
Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

1. Was hat sich in diesem Zugriffsplan auf Grund der Erstellung zusätzlicher Indizes geändert?

Das Optimierungsprogramm hat den Index genutzt, der für die Spalte JOB in der Tabelle STAFF erstellt wurde (durch eine Raute und die Bezeichnung **I_JOB** dargestellt), um diesen Zugriffsplan weiter einzuzugrenzen.



Beachten Sie im mittleren Teil des Zugriffsplandiagramms, dass für die Tabelle ORG die vorherige Indexsuche und die FETCH-Operation in die Indexsuche IXSCAN(9) geändert wurden. Durch das Hinzufügen der Spalte DEPTNAME zum Index für die Tabelle ORG kann das Optimierungsprogramm den zusätzlichen Zugriff, der die FETCH-Operation beinhaltet, ausschließen.



2. Wie effektiv ist dieser Zugriffsplan?

Dieser Zugriffsplan ist nicht so aufwändig wie der Zugriffsplan aus dem vorherigen Beispiel. Der kumulative Aufwand wurde von etwa 1.370 Timeron in Abfrage 3 auf etwa 960 Timeron in Abfrage 4 reduziert.

Weitere Schritte

Weitere Informationen zu zusätzlichen Schritten, die zur Leistungsverbesserung ausgeführt werden können, finden Sie im Handbuch *Systemverwaltung*. Sie können anschließend zu Visual Explain zurückkehren, um die Auswirkungen Ihrer Aktionen zu beurteilen.

Lerneinheit 4. Verbessern eines Zugriffsplans in einer Umgebung mit partitionierten Datenbanken

In dieser Lerneinheit erfahren Sie, wie sich der Zugriffsplan und damit verbundene Fenster für die Standardabfrage ändern, wenn Sie verschiedene Optimierungsaktivitäten ausführen. Mit Hilfe von einigen illustrierten Beispielen erfahren Sie, wie der geschätzte Gesamtaufwand für einen Zugriffsplan auch bei einfachen Abfragen verbessert werden kann, indem der Befehl **runstats** verwendet wird und entsprechende Indizes hinzugefügt werden.

Mit zunehmender Erfahrung im Umgang mit Visual Explain werden Sie noch weitere Möglichkeiten zur Optimierung von Abfragen entdecken.

Arbeiten mit Zugriffsplandiagrammen

Anhand der vier Beispiel-EXPLAIN-Momentaufnahmen werden Sie erfahren, dass die Optimierung eine wichtige Rolle bei der Datenbankleistung spielt.

Die den EXPLAIN-Momentaufnahmen zugeordneten Abfragen sind mit 1–4 numeriert. Jede Abfrage verwendet dieselbe SQL-Anweisung (wie in Lerneinheit 1 beschrieben):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Aber jede Iteration der Abfrage verwendet mehr Optimierungstechniken als die vorherige Ausführung. Bei Abfrage 1 gab es beispielsweise gar keine Leistungsverbesserung, während es bei Abfrage 4 die meisten Leistungsverbesserungen gab. Die Unterschiede bei den Abfragen werden im Folgenden beschrieben:

Abfrage 1

Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten

Abfrage 2

Erfassen aktueller Statistikdaten für die Tabellen und Indizes in einer Abfrage

Abfrage 3

Erstellen von Indizes für Spalten, die zur Verknüpfung von Tabellen in einer Abfrage verwendet werden

Abfrage 4

Erstellen zusätzlicher Indizes für Tabellenspalten

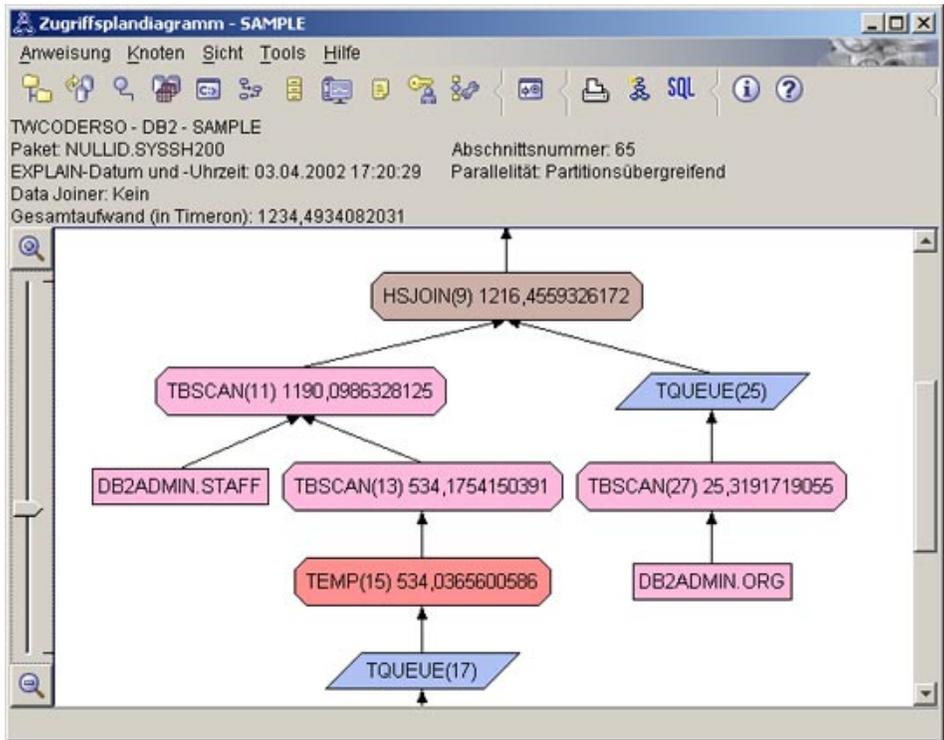
Diese Beispiele wurden auf einer RS/6000 SP-Maschine mit sieben physischen Knoten unter Verwendung der partitionsübergreifenden Parallelität generiert.

Ausführen einer Abfrage ohne Indizes und ohne Statistikdaten

In diesem Beispiel wurde der Zugriffsplan für die SQL-Abfrage ohne Indizes und ohne Statistikdaten erstellt.

Gehen Sie wie folgt vor, um das Zugriffsplandiagramm für diese Abfrage (Abfrage 1) anzuzeigen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis die Beispieldatenbank SAMPLE angezeigt wird.
2. Klicken Sie die Datenbank mit Maustaste 2 an, und wählen Sie **Protokoll mit EXPLAIN bearbeiteter Anweisungen anzeigen** aus dem Popup-Menü aus. Das Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen wird geöffnet.
3. Klicken Sie den Eintrag doppelt an, der als Abfragenummer 1 angegeben ist (eventuell müssen Sie nach rechts blättern, um die Spalte **Abfragenummer** zu finden). Das Fenster **Zugriffsplandiagramm** wird für die Anweisung geöffnet.



Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

1. Sind für jede Tabelle in der Abfrage aktuelle Statistikdaten vorhanden?

Wenn Sie überprüfen möchten, ob aktuelle Statistikdaten für jede Tabelle in der Abfrage vorhanden sind, klicken Sie jeden Tabellenknoten im Zugriffsplandiagramm doppelt an. Im entsprechenden Fenster **Tabellenstatistik**, das nun geöffnet wird, enthält die Zeile **STATS_TIME** in der Spalte **Mit EXPLAIN bearbeitet** die Worte "Statistikdaten nicht aktualisiert", die angeben, dass keine Statistikdaten zu dem Zeitpunkt erfasst waren, als die Momentaufnahme erstellt wurde.

Wenn keine aktuellen Statistikdaten vorhanden sind, verwendet das Optimierungsprogramm Standardstatistikdaten, die sich von tatsächlichen Statistikdaten unterscheiden können. Standardstatistikdaten werden mit dem Wort "Standardwert" in der Spalte **Mit EXPLAIN bearbeitet** im Fenster **Tabellenstatistik** angegeben.

Laut den Informationen im Fenster **Tabellenstatistik** für die Tabelle **ORG** hat das Optimierungsprogramm Standardstatistikdaten verwendet (wie neben den mit EXPLAIN bearbeiteten Werten angegeben).

Es wurden Standardstatistikdaten verwendet, da tatsächliche Statistikdaten nicht zur Verfügung standen, als die Momentaufnahme erstellt wurde (wie in der Zeile **STATS_TIME** angegeben).

Statistik	Mit EXPLAIN bearbeitet	Aktuell
CREATE_TIME	26.03.2002 13:35:42	
STATS_TIME	Statistikdaten nicht aktualisiert	Statistikdaten nicht aktualisiert
CARD	55(Standardwert)	
NPAGES	1(Standardwert)	
FPAGES	1(Standardwert)	
COLCOUNT	5(Standardwert)	
OVERFLOW	0(Standardwert)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
FLÜCHTIG	Kein(Standardwert)	Kein

2. Verwendet dieser Zugriffsplan die effektivsten Methoden für den Datenzugriff?

Dieser Zugriffsplan enthält Tabellensuchen anstelle von Indexsuchen. Eine Tabellensuche wird als Achteck angezeigt und mit TBSCAN angegeben. Falls eine Indexsuche verwendet worden wäre, würde sie als Raute angezeigt und mit IXSCAN angegeben. Die Verwendung eines Indexes, der für eine Tabelle erstellt wurde, ist nicht so aufwändig wie eine Tabellensuche, wenn kleine Datenmengen extrahiert werden.

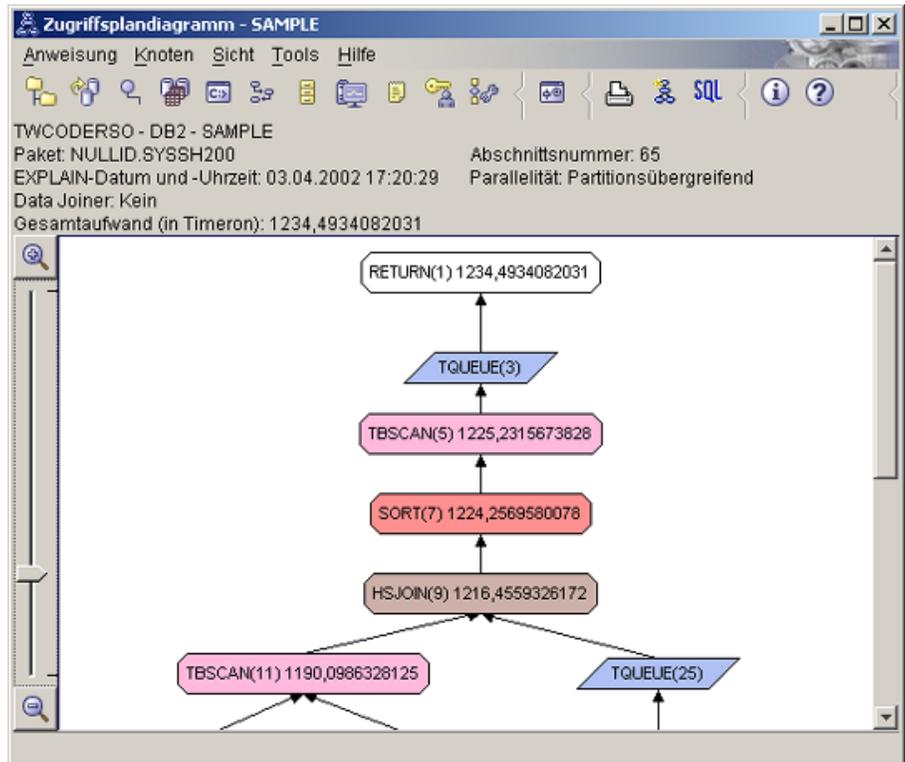
3. Wie effektiv ist dieser Zugriffsplan?

Sie können die Effektivität eines Zugriffsplans nur feststellen, wenn er auf tatsächlichen Statistikdaten basiert. Da das Optimierungsprogramm im Zugriffsplan Standardstatistikdaten verwendet hat, können Sie die Effektivität des Plans nicht feststellen.

Sie sollten sich grundsätzlich den gesamten geschätzten Aufwand für den Zugriffsplan notieren, um ihn später mit überarbeiteten Zugriffsplänen vergleichen zu können. Der in jedem Knoten aufgeführte Aufwand ist kumulativ, vom ersten Schritt Ihrer Abfrage bis einschließlich des Knotens.

Anmerkung: Für partitionierte Datenbanken ist dies der kumulative Aufwand für den Knoten, der die meisten Ressourcen verwendet.

Im Fenster **Zugriffsplandiagramm** beträgt der Gesamtaufwand etwa 1.234 Timeron, wie unter **RETURN(1)** im oberen Bereich des Diagramms angezeigt. Der gesamte geschätzte Aufwand wird ebenfalls im oberen Bereich des Fensters angezeigt.



4. Weitere Schritte

Abfrage 2 bezieht sich auf einen Zugriffspl für die Standardabfrage, nachdem der Befehl **runstats** ausgeführt wurde. Wenn Sie den Befehl **runstats** verwenden, werden dem Optimierungsprogramm die aktuellen Statistikdaten für alle Tabellen zur Verfügung gestellt, auf die die Abfrage zugreift.

Erfassen aktueller Statistikdaten für die Tabellen und Indizes mit Hilfe des Befehls 'runstats'

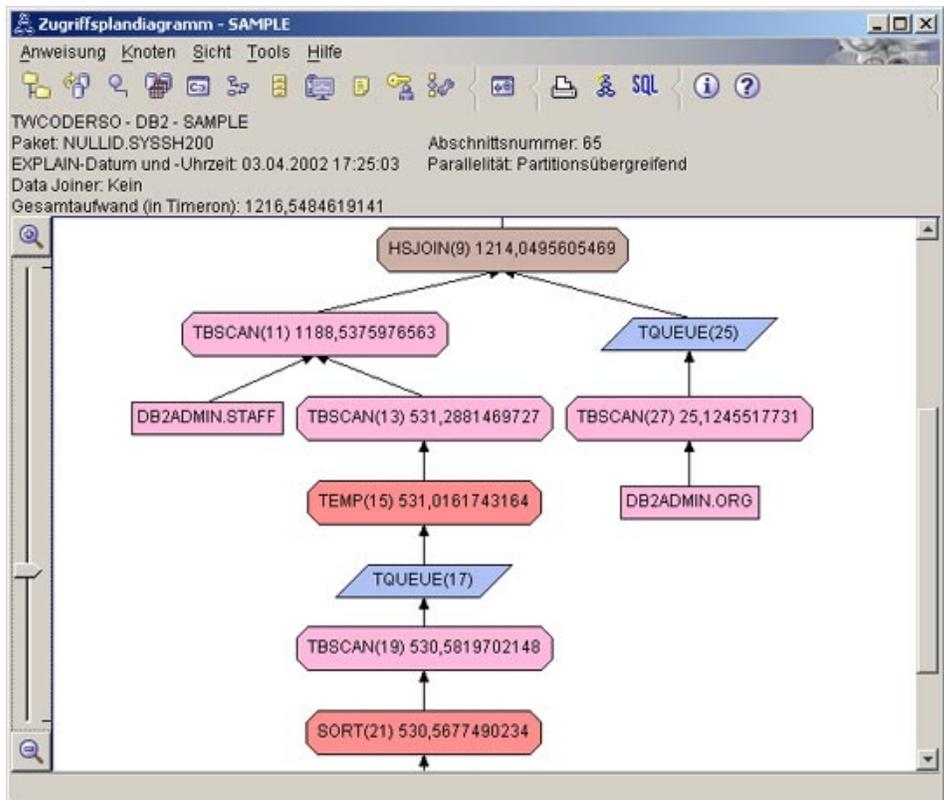
Dieses Beispiel baut auf dem Zugriffspl auf, der in Abfrage 1 beschrieben wurde, indem aktuelle Statistikdaten mit dem Befehl **runstats** erfasst werden.

Es wird dringend empfohlen, die aktuellen Statistikdaten für Tabellen und Indizes mit dem Befehl **runstats** zu erfassen, insbesondere, wenn seit der letzten Ausführung des Befehls **runstats** umfangreiche Aktualisierungen vorgenommen oder neue Indizes erstellt wurden.

Dadurch werden dem Optimierungsprogramm die genauesten Informationen zur Verfügung gestellt, anhand derer es den besten Zugriffsplan ermitteln kann. Wenn aktuelle Statistikdaten nicht verfügbar sind, kann das Optimierungsprogramm einen nicht effizienten Zugriffsplan auswählen, der auf ungenauen Standardstatistikdaten basiert.

Stellen Sie sicher, dass **runstats** nach der Aktualisierung der Tabelle ausgeführt wird. Andernfalls hält das Optimierungsprogramm die Tabelle für leer. Dies ist der Fall, wenn als Kardinalität im Fenster **Operatordetails** Null angegeben ist. In diesem Fall beenden Sie Ihre Tabellenaktualisierungen, führen den Befehl **runstats** erneut aus und erstellen erneut die EXPLAIN-Momentaufnahmen für betroffene Tabellen.

Wenn Sie das Zugriffsplandiagramm für diese Abfrage (Abfrage 2) anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 2 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.



Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

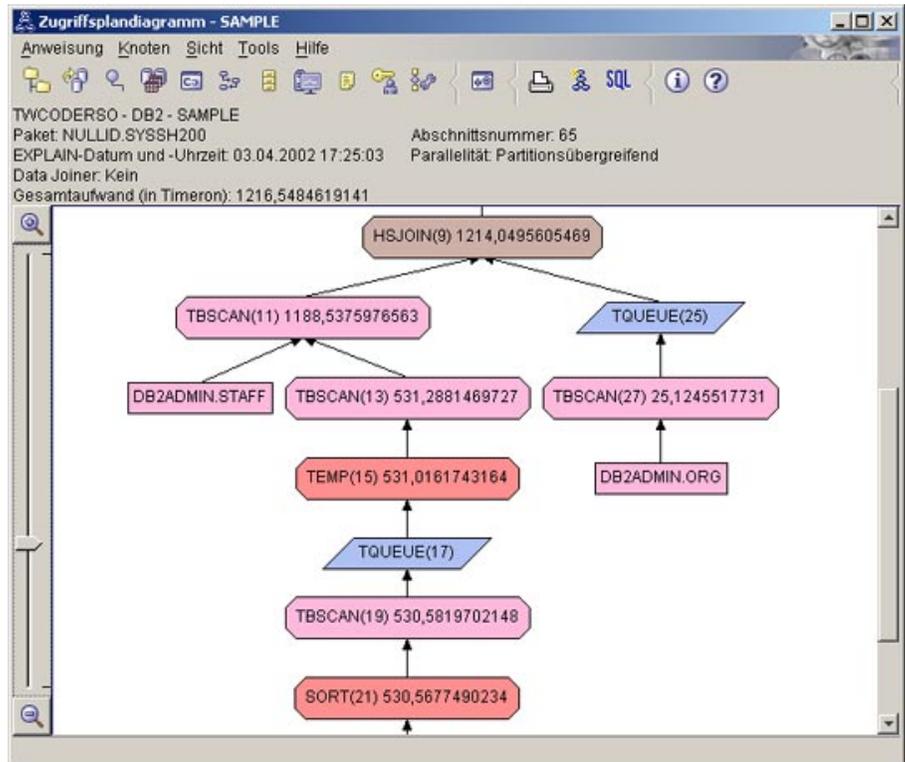
1. Sind für jede Tabelle in der Abfrage aktuelle Statistikdaten vorhanden?

Das Fenster **Tabellenstatistik** für die Tabelle ORG zeigt an, dass das Optimierungsprogramm tatsächliche Statistikdaten verwendet hat (der Wert **STATS_TIME** stellt die Zeit dar, zu der die Statistikdaten tatsächlich erfasst wurden). Die Genauigkeit der Statistikdaten hängt davon ab, ob der Inhalt der Tabellen seit der Ausführung des Befehls **runstats** erheblich geändert wurde.

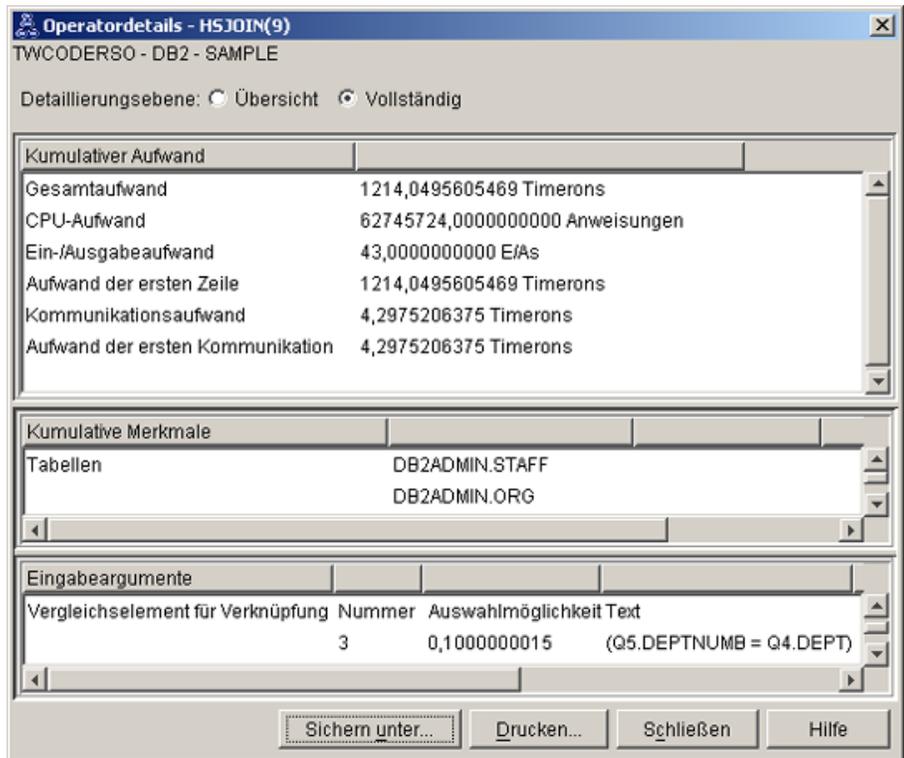
Statistik	Mit EXPLAIN bearbeitet	Aktuell
CREATE_TIME	26.03.2002 13:35:42	
STATS_TIME	03.04.2002 17:24:55	Statistikdaten nicht aktualisiert
CARD	4	
NPAGES	1	
FPAGES	1	
COLCOUNT	5	
OVERFLOW	0	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
FLÜCHTIG	Kein	Kein

2. Verwendet dieser Zugriffsplan die effektivsten Methoden für den Datenzugriff?

Wie bei Abfrage 1 verwendet der Zugriffsplan in Abfrage 2 Tabellensuchen (TBSCAN) anstelle von Indexsuchvorgängen (IXSCAN). Obwohl aktuelle Statistikdaten vorhanden sind, wurde keine Indexsuche vorgenommen, da es keine Indizes für die Spalten gibt, die von der Abfrage verwendet wurden. Die Abfrage kann verbessert werden, indem dem Optimierungsprogramm Indizes für Spalten zur Verfügung gestellt werden, die zur Verknüpfung von Tabellen verwendet werden (d. h. für Spalten, die in Verknüpfungsvergleichselementen verwendet werden). In diesem Beispiel ist dies die erste Mischsuchverknüpfung: HSJOIN(9).



Sehen Sie sich im Fenster **Operatordetails** für den Operator HSJOIN(9) den Abschnitt **Vergleichselement für Verknüpfung** unter **Eingabe-argumente** an. Die in dieser Verknüpfung verwendeten Spalten werden in der Spalte **Text** aufgelistet. In diesem Beispiel lauten diese Spalten DEPTN-UMB und DEPT.



3. Wie effektiv ist dieser Zugriffsplan?

Zugriffspläne, die auf aktuellen Statistikdaten basieren, produzieren immer einen realistischen geschätzten Aufwand (gemessen in Timeron). Da der geschätzte Aufwand in Abfrage 1 auf Standardstatistikdaten basierte, können wir den Aufwand der beiden Zugriffsplandiagramme nicht vergleichen, um festzustellen, welcher effektiver ist. Es spielt keine Rolle, ob der Aufwand höher oder niedriger ist. Sie müssen den Aufwand von Zugriffsplänen vergleichen, die auf tatsächlichen Statistikdaten basieren, um ein gültiges Maß für die Effektivität zu erhalten.

4. Weitere Schritte

Abfrage 3 bezieht sich darauf, welche Auswirkungen das Hinzufügen von Indizes für die Spalten DEPTNUMB und DEPT hat. Wenn Sie Indizes für die Spalten hinzufügen, die in Verknüpfungsvergleichselementen verwendet werden, kann dies die Leistung erhöhen.

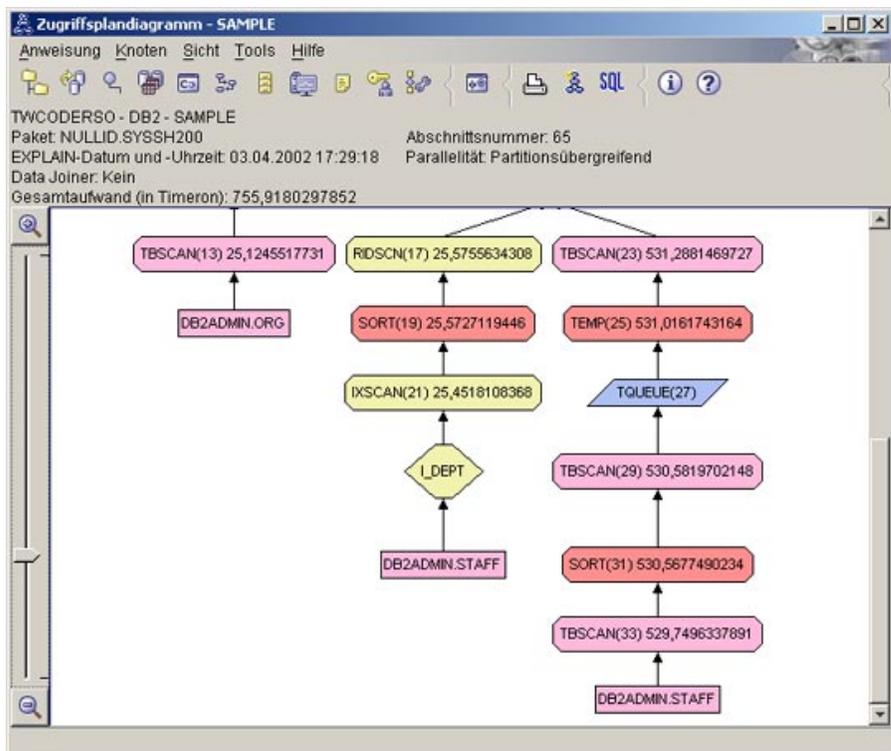
Erstellen von Indizes für Spalten, die verwendet werden, um Tabellen in einer Abfrage zu verknüpfen

Dieses Beispiel baut auf dem Zugriffsplan auf, der in Abfrage 2 beschrieben wurde, indem Indizes für die Spalte DEPT in der Tabelle STAFF und für die Spalte DEPTNUMB in der Tabelle ORG erstellt werden.

Anmerkung: In Version 8 können empfohlene Indizes mit Hilfe des Assistenten für die Auslastungsleistung erstellt werden.

Wenn Sie das Zugriffsplandiagramm für diese Abfrage (Abfrage 3) anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 3 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.

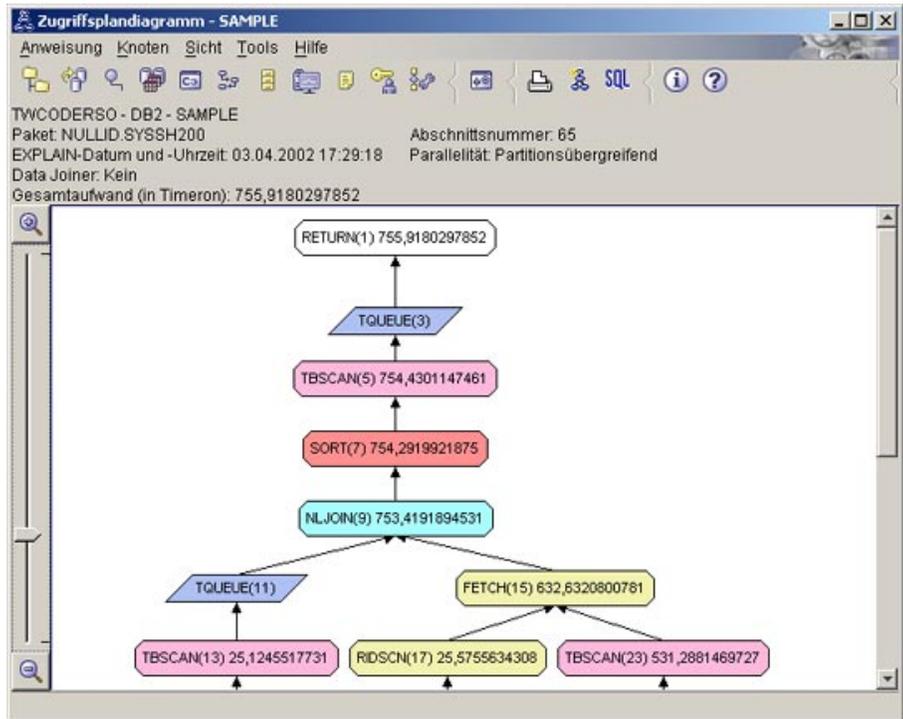
Anmerkung: Obwohl ein Index für DEPTNUM erstellt wurde, wurde er vom Optimierungsprogramm nicht verwendet.



Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

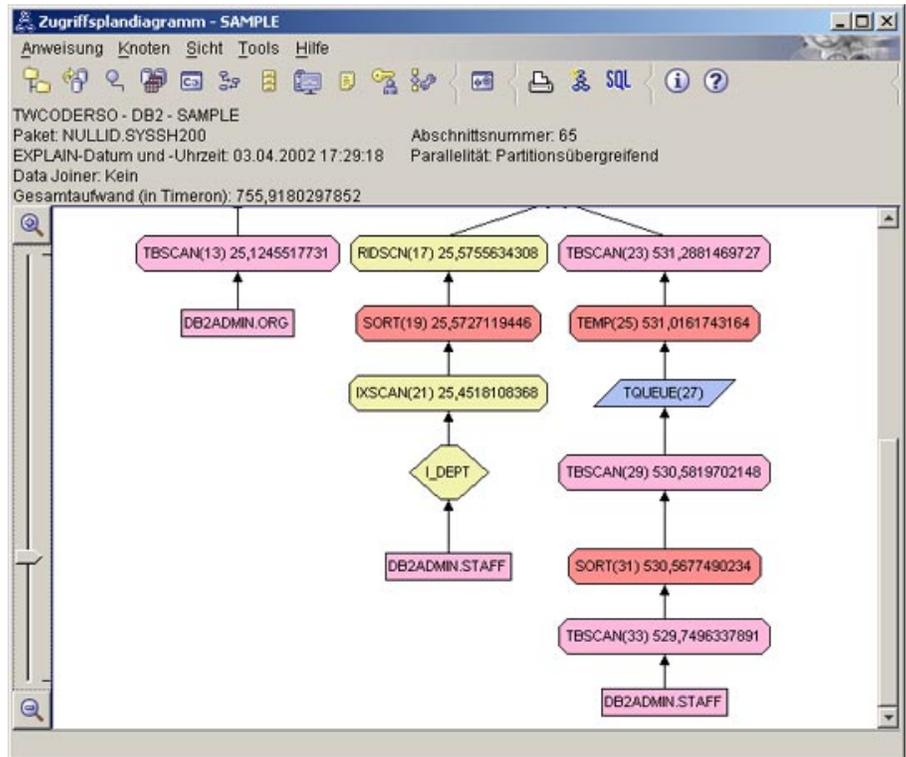
1. Was hat sich im Zugriffsplan mit Indizes geändert?

Es wurde ein neuer Knoten, **I_DEPT**, in Form einer Raute über der Tabelle STAFF hinzugefügt. Dieser Knoten stellt den Index dar, der für DEPT erstellt wurde. Er zeigt an, dass das Optimierungsprogramm eine Indexsuche anstelle einer Tabellensuche verwendet hat, um festzustellen, welche Zeilen abgerufen werden müssen.



2. Verwendet dieser Zugriffsplan die effektivsten Methoden für den Datenzugriff?

Der Zugriffsplan für diese Abfrage zeigt an, welche Auswirkungen das Erstellen von Indizes für die Spalte DEPTNUMB der Tabelle ORG hat, was zu FETCH(15) und IXSCAN(21) führte, sowie die Auswirkungen auf die Spalte DEPT der Tabelle STAFF. Da es in Abfrage 2 diesen Index nicht gab, wurde dort eine Tabellensuche verwendet.



Das Fenster **Operatordetails** für den Operator FETCH(15) zeigt die Spalten an, die in dieser Operation verwendet werden.

Kumulativer Aufwand	
Gesamtaufwand	632,6320800781 Timerons
CPU-Aufwand	9049302,0000000000 Anweisungen
Ein-/Ausgabeaufwand	37,9756927490 E/As
Aufwand der ersten Zeile	582,9780883789 Timerons
Kommunikationsaufwand	4,8595042229 Timerons
Aufwand der ersten Kommunikation	0,0000000000 Timerons

Kumulative Merkmale	
Tabellen	DB2ADMIN.STAFF
Spalten	DB2ADMIN.STAFF.\$RID\$
	DB2ADMIN.STAFF.NAME

Eingabeargumente	
Abgerufene Spalten	DB2ADMIN.STAFF.NAME
	DB2ADMIN.STAFF.ID
	DB2ADMIN.STAFF.COMM

Die Kombination von Indexsuche und Abruf wird als weniger aufwändig berechnet als die vollständige Tabellensuche, die in den vorherigen Zugriffsplänen verwendet wurde.

3. Wie effektiv ist dieser Zugriffsplan?

Dieser Zugriffsplan ist nicht so aufwändig wie der Zugriffsplan aus dem vorherigen Beispiel. Der kumulative Aufwand wurde von etwa 1.214 Timeron in Abfrage 2 auf etwa 755 Timeron in Abfrage 3 reduziert.

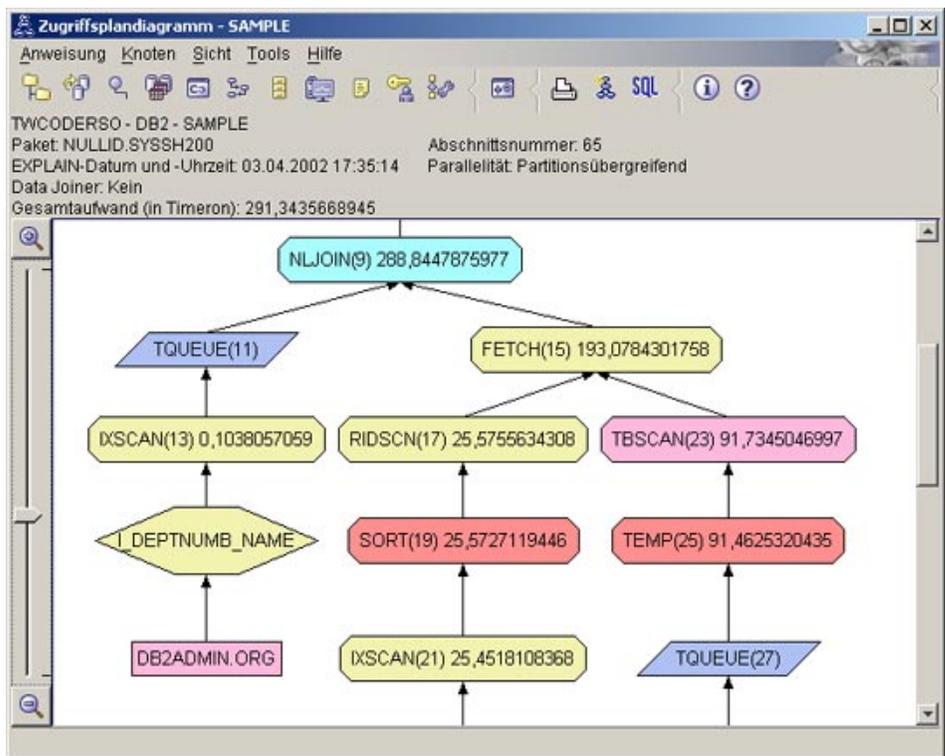
4. Weitere Schritte

Abfrage 4 reduziert die FETCH-Operation und die Indexsuche auf eine einzige Indexsuche ohne FETCH-Operation. Durch das Erstellen zusätzlicher Indizes kann der geschätzte Aufwand für den Zugriffsplan reduziert werden.

Erstellen zusätzlicher Indizes für Tabellenspalten

Dieses Beispiel baut auf dem Zugriffsplan auf, der in Abfrage 3 beschrieben wurde, indem ein Index für die Spalte JOB in der Tabelle STAFF erstellt wird und indem DEPTNAME zum vorhandenen Index in der Tabelle ORG hinzugefügt wird. (Das Hinzufügen eines separaten Indexes könnte zu einem zusätzlichen Zugriff führen.)

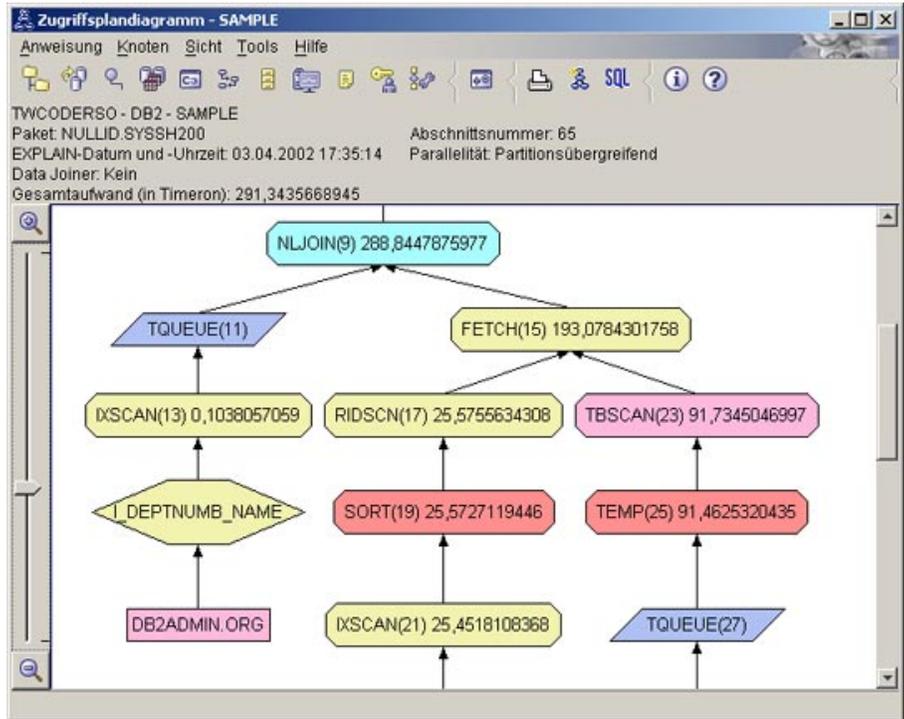
Wenn Sie das Zugriffsplandiagramm für diese Abfrage anzeigen möchten, klicken Sie im Fenster mit dem Protokoll der mit EXPLAIN bearbeiteten Anweisungen den Eintrag doppelt an, der als Abfragenummer 4 angegeben ist. Das Fenster **Zugriffsplandiagramm** für diese Ausführung der Anweisung wird geöffnet.



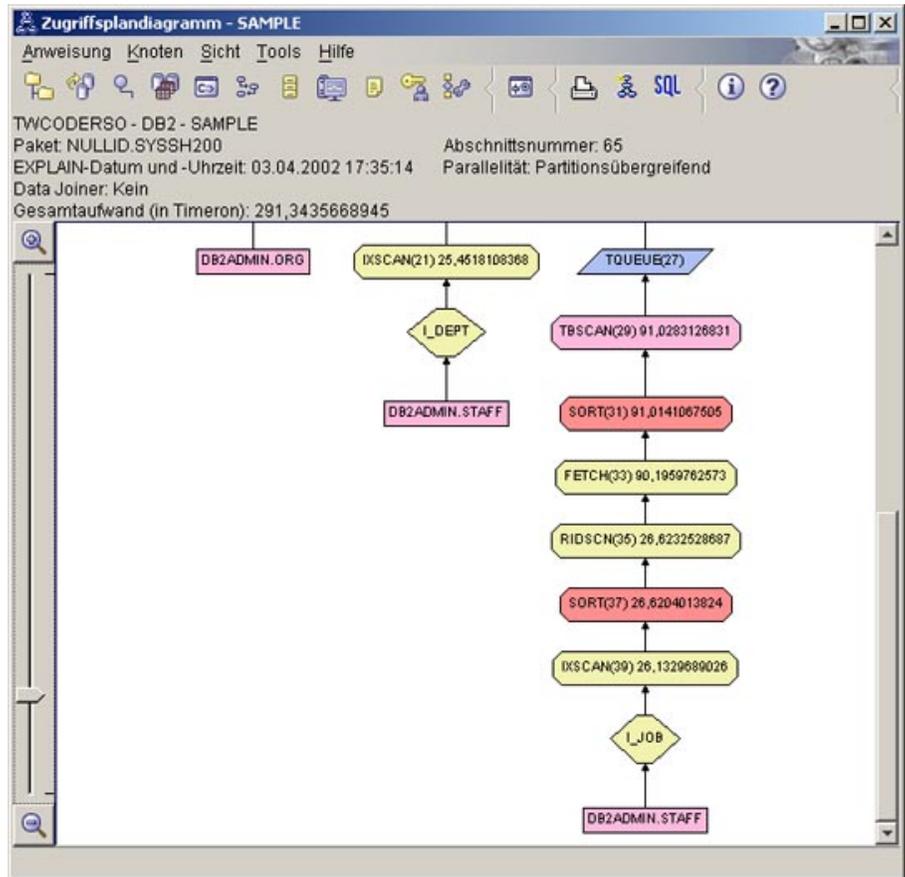
Durch Beantwortung der folgenden Fragen erhalten Sie Hinweise dazu, wie Sie die Abfrage verbessern können.

1. Was hat sich in diesem Zugriffsplan auf Grund der Erstellung zusätzlicher Indizes geändert?

Beachten Sie im mittleren Teil des Zugriffsplandiagramms, dass für die Tabelle ORG die vorherige Tabellensuche in die Indexsuche IXSCAN(7) geändert wurde. Durch das Hinzufügen der Spalte DEPTNAME zum Index für die Tabelle ORG kann das Optimierungsprogramm den Zugriff eingrenzen, der die Tabellensuche beinhaltet.



Beachten Sie im unteren Teil des Zugriffsplandiagramms, dass für die Tabelle STAFF die vorherige Indexsuche und die FETCH-Operation in die Indexsuche IXSCAN(39) geändert wurden. Durch das Erstellen des Indexes JOB für die Tabelle STAFF kann das Optimierungsprogramm den zusätzlichen Zugriff, der die FETCH-Operation beinhaltet, ausschließen.



2. Wie effektiv ist dieser Zugriffsplan?

Dieser Zugriffsplan ist nicht so aufwändig wie der Zugriffsplan aus dem vorigen Beispiel. Der kumulative Aufwand wurde von etwa 753 Timeron in Abfrage 3 auf etwa 290 Timeron in Abfrage 4 reduziert.

Weitere Schritte

Weitere Informationen zu zusätzlichen Schritten, die zur Leistungsverbesserung ausgeführt werden können, finden Sie im Handbuch *Systemverwaltung*. Sie können anschließend zu Visual Explain zurückkehren, um die Auswirkungen Ihrer Aktionen zu beurteilen.

Anhang A. Visual Explain-Konzepte

Abfrageoptimierungsklasse

Eine *Abfrageoptimierungsklasse* besteht aus einer Reihe von Regeln für das Umschreiben einer Abfrage und von Optimierungstechniken für das Kompilieren von Abfragen.

Die primären Abfrageoptimierungsklassen lauten:

- 1 Eingeschränkte Optimierung. Sinnvoll, wenn die Speicherkapazität und die Verarbeitungsressourcen erheblich eingeschränkt sind. Entspricht in etwa der Optimierung aus Version 1.
- 2 Geringe Optimierung. Gibt eine Optimierungsstufe höher als die aus Version 1 an, jedoch mit wesentlich geringerem Optimierungsaufwand als Stufe 3 und höher, insbesondere bei sehr komplexen Abfragen.
- 3 Mäßige Optimierung. Entspricht am ehesten der Abfrageoptimierungskenndaten von DB2 für MVS/ESA.
- 5 Normale Optimierung. Wird für eine heterogene Umgebung empfohlen, die sowohl einfache Transaktionen als auch komplexe Abfragen umfasst.
- 7 Normale Optimierung. Mit Abfrageoptimierung 5 identisch, mit der Ausnahme, dass das Maß der Abfrageoptimierung für komplexe dynamische SQL-Abfragen nicht verringert wird.

Weitere Abfrageoptimierungsklassen, die nur unter besonderen Umständen verwendet werden, lauten:

- 0 Minimale Optimierung. Verwenden Sie diese Optimierung nur, wenn eine geringe oder gar keine Optimierung erforderlich ist (d. h. für sehr einfache Abfragen in gut indextierten Tabellen).
- 9 Maximale Optimierung. Verwendet erhebliche Speicherkapazität und Verarbeitungsressourcen. Verwenden Sie diese Optimierung nur, wenn Klasse 5 nicht ausreicht (d. h. für sehr komplexe Abfragen, die sehr lange ausgeführt werden und die mit Klasse 5 nicht erfolgreich ausgeführt werden).

Allgemein sollten Sie eine höhere Optimierungsstufe für statische Abfragen und für Abfragen verwenden, von denen Sie erwarten, dass sie eine längere Zeit für die Ausführung beanspruchen. Sie sollten eine geringere Optimierungsstufe für einfache Abfragen verwenden, die dynamisch übergeben werden oder die nur wenige Male ausgeführt werden.

Geben Sie den folgenden Befehl im Befehlszeilenprozessor ein, um die Abfrageoptimierung für dynamische SQL-Anweisungen festzulegen:

```
SET CURRENT QUERY OPTIMIZATION = n;
```

Dabei ist 'n' die gewünschte Abfrageoptimierungsklasse.

Verwenden Sie die Option `QUERYOPT` des Befehls **BIND** oder **PREP**, um die Abfrageoptimierung für statische SQL-Anweisungen festzulegen.

Weitere Informationen finden Sie im Abschnitt zur Anpassung der Optimierungsklasse im Handbuch *Systemverwaltung*.

Aufwand

Unter *Aufwand* versteht man im Zusammenhang mit Visual Explain die gesamte geschätzte Ressourcennutzung, die notwendig ist, um den Zugriffsplan für eine Anweisung (oder die Elemente einer Anweisung) auszuführen. Der Aufwand wird aus einer Kombination des CPU-Aufwands (Anzahl der Anweisungen) und E/A (Anzahl der Suchvorgänge und Seitenübertragungen) abgeleitet.

Die Einheit für den Aufwand lautet *Timeron*. Ein Timeron entspricht nicht direkt der tatsächlich abgelaufenen Zeit, gibt jedoch eine grobe relative Schätzung der Ressourcen (Aufwand) an, die der Datenbankmanager benötigt, um zwei Pläne für dieselbe Abfrage auszuführen.

Der Aufwand, der in jedem Operatorknoten eines Zugriffsplandiagramms angezeigt wird, ist der kumulative Aufwand von Beginn der Ausführung des Zugriffsplans bis einschließlich zur Ausführung dieses bestimmten Operators. Es werden keine Faktoren wie die Auslastung des Systems oder der Aufwand für das Zurückgeben von Datenzeilen an den Benutzer berücksichtigt.

Behälter

Ein *Behälter* ist eine physische Speicherposition für die Daten. Er ist einem Tabellenbereich zugeordnet und kann eine Datei, ein Verzeichnis oder eine Einheit sein.

Clustering

Mit der Zeit können Aktualisierungen dazu führen, dass Zeilen auf Datenseiten die Position ändern, wodurch der Grad der *Clusterbildung* verringert wird, der zwischen einem Index und den Datenseiten vorhanden ist. Wenn Sie eine Tabelle hinsichtlich eines ausgewählten Index reorganisieren, werden die Daten erneut in Gruppen zusammengefasst. Ein in Gruppen zusammengefasster Index ist besonders sinnvoll für Spalten, die über Bereichsvergleichs-

elemente verfügen, da er einen besseren sequenziellen Zugriff auf Daten in der Basistabelle ermöglicht. Dies führt zu weniger Seitenabrufen, da sich ähnliche Werte auf derselben Datenseite befinden.

Allgemein gilt, dass nur einer der Indizes in einer Tabelle einen hohen Grad an Clusterbildung aufweisen kann.

Wenn Sie den Grad der Clusterbildung für einen Index prüfen möchten, klicken Sie seinen Knoten doppelt an, um das Fenster **Indexstatistik** anzuzeigen. In diesem Fenster werden die Werte für das Clusterverhältnis oder den Clusterfaktor angezeigt. Wenn der Wert niedrig ist, sollten Sie die Daten der Tabelle reorganisieren.

Weitere Informationen finden Sie im Abschnitt zur Reorganisation von Tabellendaten im Handbuch *Systemverwaltung*.

Cursorblockung

Die *Cursorblockung* ist eine Technik, mit der der Systemaufwand verringert wird, indem der Datenbankmanager einen *Block* von Zeilen in einer einzigen Operation abruft. Diese Zeilen werden in einem Cache gespeichert, während sie verarbeitet werden. Der Cache wird zugeordnet, wenn eine Anwendung die Anforderung OPEN CURSOR absetzt. Die Zuordnung wird zurückgenommen, wenn der Cursor geschlossen wird. Wenn alle Zeilen verarbeitet wurden, wird ein weiterer Block mit Zeilen abgerufen.

Verwenden Sie die Option BLOCKING der Befehle **PREP** oder **BIND** mit den folgenden Parametern, um den Typ der Cursorblockung anzugeben:

UNAMBIG

Es werden nur eindeutige Cursor geblockt (Standardwert).

ALL Es werden sowohl mehrdeutige als auch eindeutige Cursor geblockt.

NO Es werden keine Cursor geblockt.

Weitere Informationen finden Sie im Abschnitt zur Cursorblockung im Handbuch *Systemverwaltung*.

Dynamisches SQL

Dynamische SQL-Anweisungen sind SQL-Anweisungen, die innerhalb eines Anwendungsprogramms vorbereitet und ausgeführt werden, während das Programm ausgeführt wird. Für dynamisches SQL gilt eine der folgenden Möglichkeiten:

- Sie setzen die SQL-Anweisung interaktiv über die Befehlszeilenschnittstelle oder den Befehlszeilenprozessor ab.

- Die SQL-Quelle ist in Hostprogrammiersprachenvariablen enthalten, die in ein Anwendungsprogramm eingebettet sind.

Wenn DB2 eine dynamische SQL-Anweisung ausführt, wird ein Zugriffsplan erstellt, der auf den aktuellen Katalogstatistiken und Konfigurationsparametern basiert. Dieser Zugriffsplan kann sich von einer Ausführung der Anweisungen im Anwendungsprogramm bis zur nächsten Ausführung ändern.

Die Alternative zum dynamischen SQL ist statisches SQL.

EXPLAIN-Momentaufnahme

Mit Visual Explain können Sie den Inhalt einer EXPLAIN-Momentaufnahme untersuchen.

Eine *EXPLAIN-Momentaufnahme* besteht aus komprimierten Informationen, die erfasst werden, wenn eine SQL-Anweisung mit EXPLAIN bearbeitet wird. Sie wird als großes Binärobjekt (BLOB) in der Tabelle EXPLAIN_STATEMENT gespeichert, und enthält die folgenden Informationen:

- Die interne Darstellung des Zugriffsplans, einschließlich seiner Operatoren und der Tabellen und Indizes, auf die zugegriffen wird.
- Die Entscheidungsbedingungen, die vom Optimierungsprogramm verwendet werden, einschließlich der Statistikdaten für Datenbankobjekte und des kumulativen Aufwands für jede Operation.

Eine EXPLAIN-Momentaufnahme ist erforderlich, wenn Sie die grafische Darstellung des Zugriffsplans einer SQL-Anweisung anzeigen möchten. Wenn Sie sicherstellen möchten, dass eine EXPLAIN-Momentaufnahme erstellt wird, muss Folgendes beachtet werden:

1. EXPLAIN-Tabellen müssen im Datenbankmanager vorhanden sein, damit die EXPLAIN-Momentaufnahmen gespeichert werden können. Weitere Informationen zur Erstellung dieser Tabellen finden Sie im Abschnitt zum Erstellen der EXPLAIN-Tabellen in der Onlinehilfefunktion.
2. Setzen Sie die Option EXPLSNAP für ein Paket, das statische SQL-Anweisungen enthält, auf ALL oder YES, wenn Sie das Paket vorbereiten oder binden. Sie erhalten eine EXPLAIN-Momentaufnahme für jede mit EXPLAIN bearbeitbare SQL-Anweisung im Paket. Weitere Informationen zu den Befehlen **BIND** und **PREP** finden Sie im Handbuch *Command Reference*.
3. Setzen Sie die Option EXPLSNAP für dynamische SQL-Anweisungen auf ALL, wenn Sie die Anwendung binden, die die Anweisungen absetzt, oder setzen Sie das Sonderregister CURRENT EXPLAIN SNAPSHOT auf YES

oder EXPLAIN, bevor Sie die Anweisungen interaktiv absetzen. Weitere Informationen finden Sie im Abschnitt zu aktuellen EXPLAIN-Momentaufnahmen im Handbuch *SQL Reference*.

Mit EXPLAIN bearbeitbare Anweisung

Eine *mit EXPLAIN bearbeitbare Anweisung* ist eine SQL-Anweisung, für die eine EXPLAIN-Operation ausgeführt werden kann.

Zu den mit EXPLAIN bearbeitbaren SQL-Anweisungen gehören:

- SELECT
- INSERT
- UPDATE
- DELETE
- VALUES

Mit EXPLAIN bearbeitete Anweisung

Eine *mit EXPLAIN bearbeitete Anweisung* ist eine SQL-Anweisung, für die eine EXPLAIN-Operation ausgeführt wurde. Mit EXPLAIN bearbeitete Anweisungen werden im Fenster mit dem Protokoll mit EXPLAIN bearbeiteter Anweisungen angezeigt.

Operand

Ein Operand ist eine Entität, für die eine Operation ausgeführt wird. Beispiel: Eine Tabelle oder ein Index ist ein Operand verschiedener Operatoren wie TBSCAN und IXSCAN.

Operator

Ein *Operator* ist entweder eine Aktion, die für Daten ausgeführt werden muss, oder die Ausgabe aus einer Tabelle oder einem Index, wenn der Zugriffsplan für eine SQL-Anweisung ausgeführt wird.

Die folgenden Operatoren können im Zugriffsplandiagramm vorkommen:

DELETE

Löscht Zeilen aus einer Tabelle.

EISCAN

Durchsucht einen benutzerdefinierten Index, um einen verringerten Datenstrom von Zeilen zu produzieren.

FETCH

Ruft Spalten aus einer Tabelle über eine spezielle Satzkenung ab.

FILTER

Filtert Daten, indem mindestens ein Vergleichselement auf die Daten angewendet wird.

GRPBY

Gruppiert Zeilen nach gemeinsamen Werten von zugewiesenen Spalten oder Funktionen und wertet festgelegte Funktionen aus.

HSJOIN

Stellt eine Hashverknüpfung dar, bei der mindestens zwei Tabellen in den verknüpften Spalten im Hashverfahren verknüpft werden.

INSERT

Fügt Zeilen in eine Tabelle ein.

IXAND

Fügt die Zeilenkennungen (Satz-IDs) aus mindestens zwei Indexsuchen über AND zusammen.

IXSCAN

Durchsucht einen Index einer Tabelle mit optionalen Start/Stop-Bedingungen, wobei ein geordneter Datenstrom von Zeilen produziert wird.

MSJOIN

Stellt eine Mischverknüpfung dar, bei der äußere und innere Tabellen die Reihenfolge des Verknüpfungsvergleichselements aufweisen müssen.

NLJOIN

Stellt eine verschachtelte Schleifenverknüpfung dar, die auf eine innere Tabelle einmal für jede Zeile der äußeren Tabelle zugreift.

RETURN

Stellt die Rückgabe der Daten aus der Abfrage an den Benutzer dar.

RIDSCAN

Durchsucht eine Liste von Zeilenkennungen (Satz-IDs), die aus mindestens einem Index erhalten wurden.

SHIP

Ruft Daten aus einer fernen Datenbankquelle ab. Wird im System zusammengeschlossener Datenbanken verwendet.

SORT

Sortiert Zeilen in der Reihenfolge der angegebenen Spalten und schließt optional doppelte Einträge aus.

TBSCAN

Ruft Zeilen ab, indem alle erforderlichen Daten direkt aus den Daten-seiten gelesen werden.

TEMP

Speichert Daten in einer temporären Tabelle, damit sie (möglicherweise mehrfach) wieder aus dieser Tabelle gelesen werden können.

TQUEUE

Überträgt Tabellendaten zwischen Datenbankagenten.

UNION

Verkettet Datenströme von Zeilen aus mehreren Tabellen.

UNIQUE

Schließt Zeilen mit doppelten Werten für angegebene Spalten aus.

UPDATE

Aktualisiert Zeilen in einer Tabelle.

CMPEXP

Operatorname: CMPEXP

Stellt Folgendes dar: Die Verarbeitung von Ausdrücken, die für temporäre oder endgültige Ergebnisse erforderlich sind.

(Dieser Operator gilt nur für den Debugmodus.)

DELETE

Operatorname: DELETE

Stellt Folgendes dar: Das Löschen von Zeilen aus einer Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffspaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu löschenden Zeilensatz definieren.

Vorschlag zur Leistungsverbesserung:

- Wenn Sie alle Zeilen aus einer Tabelle löschen, sollten Sie die Anweisung DROP TABLE oder den Befehl **LOAD REPLACE** verwenden.

EISCAN

Operatorname: EISCAN

Stellt Folgendes dar: Dieser Operator durchsucht einen benutzerdefinierten Index, um einen verringerten Datenstrom von Zeilen zu produzieren.

Beim Durchsuchen werden die mehrfachen Start/Stop-Bedingungen aus der Bereichsgenerierungsfunktion verwendet, die vom Benutzer eingerichtet wurde.

Diese Operation wird ausgeführt, um den Satz qualifizierender Zeilen einzuschränken, bevor auf die Basistabelle (basierend auf Vergleichselementen) zugegriffen wird.

Vorschlag zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Indexes korrigiert werden.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

FETCH

Operatorname: FETCH

Stellt Folgendes dar: Das Abrufen von Spalten aus einer Tabelle mit Hilfe einer speziellen Zeilenkennung (RID).

Vorschläge zur Leistungsverbesserung:

- Erweitern Sie die Indexschlüssel, um die abgerufenen Spalten aufzunehmen, so dass auf die Datenseiten nicht zugegriffen werden muss.
- Suchen Sie den Index, der mit dem Abruf verbunden ist, und klicken Sie seinen Knoten doppelt an, um das Fenster mit den Statistikdaten anzuzeigen. Stellen Sie sicher, dass der Grad der Clusterbildung für den Index hoch ist.
- Erhöhen Sie die Puffergröße, wenn die Eingabe/Ausgabe (E/A) bei dem Abruf größer als die Anzahl der Seiten in der Tabelle ist.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Die Statistiken zum Quantil und zu den häufigen Werten bieten Informationen zur Selektivität von Vergleichselementen, die bestimmen, wann eine Indexsuche gegenüber einer Tabellensuche bevorzugt wird. Verwenden Sie den Befehl **runstats** für eine Tabelle mit der Klausel WITH DISTRIBUTION, um diese Statistikdaten zu aktualisieren.

FILTER

Operatorname: FILTER

Stellt Folgendes dar: Die Anwendung von verbleibenden Vergleichselementen, so dass Daten auf der Basis der Bedingungen der Vergleichselemente gefiltert werden.

Vorschläge zur Leistungsverbesserung:

- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.
- Stellen Sie sicher, dass die Optimierungsklasse mindestens 3 beträgt, so dass das Optimierungsprogramm eine Verknüpfung anstelle einer Unterabfrage verwendet. Wenn dies nicht möglich ist, sollten Sie versuchen, die SQL-Abfrage selbst umzuschreiben, um die Unterabfrage auszuschließen. Ein Beispiel finden Sie im Abschnitt zum Umschreiben von Abfragen durch den SQL-Compiler im Handbuch *Systemverwaltung*.

GENROW

Operatorname: GENROW

Stellt Folgendes dar: Eine integrierte Funktion, die eine Tabelle mit Zeilen generiert, wobei keine Eingaben aus Tabellen, Indizes oder Operatoren verwendet werden.

GENROW kann vom Optimierungsprogramm verwendet werden, um Datenzeilen zu generieren (z. B. für die Anweisung INSERT oder für einige Listen IN, die in Verknüpfungen umgesetzt werden).

Wenn Sie die geschätzten Statistikdaten für die von der Funktion GENROW generierten Tabellen anzeigen möchten, klicken Sie den Knoten dieser Funktion doppelt an.

GRPBY

Operatorname: GRPBY

Stellt Folgendes dar: Das Gruppieren von Zeilen nach gemeinsamen Werten zugewiesener Spalten oder Funktionen. Diese Operation ist erforderlich, um eine Gruppe von Werten herzustellen oder um festgelegte Funktionen auszuwerten.

Wenn keine GROUP BY-Spalten angegeben werden, kann der Operator GRPBY dennoch verwendet werden, wenn es Spaltenfunktionen in der SELECT-Liste gibt, die angeben, dass die gesamte Tabelle als einzelne Gruppe behandelt werden soll, wenn die Spaltenberechnung vorgenommen wird.

Vorschläge zur Leistungsverbesserung:

- Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsplanaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu gruppierenden Zeilensatz definieren.
- Gehen Sie wie folgt vor, um die Leistung der Anweisung SELECT zu verbessern, die eine Spaltenfunktion, jedoch keine GROUP BY-Klausel enthält:
 - Erstellen Sie für die Spaltenfunktion MIN(C) einen aufsteigenden Index für C.
 - Erstellen Sie für die Spaltenfunktion MAX(C) einen absteigenden Index für C.

HSJOIN

Operatorname: HSJOIN

Stellt Folgendes dar: Eine Hashverknüpfung, für die die qualifizierten Zeilen aus Tabellen im Hashverfahren verknüpft werden, um eine direkte Verknüpfung zu ermöglichen, ohne dass der Inhalt der Tabellen zuvor geordnet wird.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine Hashverknüpfung ist möglich, wenn es ein Verknüpfungsvergleichselement gibt, das die Spalten aus zwei verschiedenen Tabellen gleichsetzt. Die Verknüpfungsvergleichselemente müssen genau denselben Datentyp aufweisen. Hashverknüpfungen können auch aus umgeschriebenen Unterabfragen entstehen, wie dies bei NLJOIN der Fall ist.

Bei einer Hashverknüpfung müssen die Eingabetabellen nicht geordnet sein. Die Verknüpfung wird vorgenommen, indem die innere Tabelle der Hashverknüpfung durchsucht wird und eine Referenztable generiert wird, indem die zu verknüpfenden Spaltenwerte im Hashverfahren verknüpft werden. Anschließend wird die äußere Tabelle gelesen, wobei die zu verknüpfenden Spaltenwerte im Hashverfahren verknüpft werden und in der Referenztable, die für die innere Tabelle generiert wurde, überprüft werden.

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Verwenden Sie lokale Vergleichselemente (d. h. Vergleichselemente, die nur auf eine Tabelle verweisen), um die Anzahl der zu verknüpfenden Zeilen zu verringern.
- Erhöhen Sie die Größe des Zwischenspeichers für Sortierlisten, damit die Hashreferenztable im Speicher gehalten werden kann.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

INSERT

Operatorname: INSERT

Stellt Folgendes dar: Das Einfügen von Zeilen in eine Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffspaufwand verbessern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den einzufügenden Zeilensatz definieren.

IXAND

Operatorname: IXAND

Stellt Folgendes dar: Das logische Verknüpfen der Ergebnisse von Mehrfachindexsuchen mit AND und Techniken für die dynamische Bitzuordnung. Der Operator ermöglicht es, dass Vergleichselemente, die über AND verknüpft sind, auf mehrere Indizes angewendet werden, um die Zugriffe auf zugrundeliegende Tabellen auf ein Minimum zu reduzieren.

Dieser Operator wird aus folgenden Gründen ausgeführt:

- Einschränken des Zeilensatzes, bevor auf die Basistabelle zugegriffen wird
- Verknüpfen von Vergleichselementen über AND, die auf mehrere Indizes angewendet werden
- Verknüpfen von Ergebnissen aus einfachen Gleichheitsverknüpfungen über AND, die in Sternverknüpfungen verwendet werden

Vorschläge zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Indexes korrigiert werden.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.
- Im Allgemeinen ist eine Indexsuche am effektivsten, wenn sich nur wenige Zeilen qualifizieren. Das Optimierungsprogramm verwendet die Statistikdaten, die für die in den Vergleichselementen verwiesenen Spalten verfügbar sind, um die Anzahl der sich qualifizierenden Zeilen zu schätzen. Wenn einige Werte häufiger auftreten als andere, sollten Sie unbedingt eine Verteilungstatistik anfordern, indem Sie die Klausel WITH DISTRIBUTION für den Befehl **runstats** verwenden. Mit Hilfe der nicht einheitlichen

Verteilungsstatistik kann das Optimierungsprogramm zwischen häufig auftretenden und weniger häufig auftretenden Werten unterscheiden.

- IXAND kann am besten für einspaltige Indizes genutzt werden, da Start- und Stoppschlüssel bei der Verwendung von IXAND kritisch sind.
- Für Sternverknüpfungen erstellen Sie einspaltige Indizes für die Spalten mit der höchsten Selektivität in der Faktabelle und den dazugehörigen Dimensionstabellen.

IXSCAN

Operatorname: IXSCAN

Stellt Folgendes dar: Das Durchsuchen eines Indexes, um einen verringerten Datenstrom von Zeilen zu produzieren. Beim Durchsuchen können optionale Start/Stop-Bedingungen verwendet werden oder indexierbare Vergleichselemente angewendet werden, die auf Spalten des Indexes verweisen.

Diese Operation wird ausgeführt, um den Satz qualifizierender Zeilen einzuschränken, bevor auf die Basistabelle (basierend auf Vergleichselementen) zugegriffen wird.

Weitere Informationen finden Sie im Abschnitt zur Indexsuche im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Index korrigiert werden.
- Wenn auf mindestens zwei Tabellen zugegriffen wird, kann ein Zugriff auf die innere Tabelle über einen Index effektiver werden, wenn ein Index in der verknüpften Spalte der äußeren Tabelle eingerichtet wird.

Weitere Richtlinien zu Indizes finden Sie in der Onlinehilfefunktion für Visual Explain.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.
- Im Allgemeinen ist eine Indexsuche am effektivsten, wenn sich nur wenige Zeilen qualifizieren. Das Optimierungsprogramm verwendet die Statistikdaten, die für die in den Vergleichselementen verwiesenen Spalten verfügbar sind, um die Anzahl der sich qualifizierenden Zeilen zu schätzen. Wenn einige Werte häufiger auftreten als andere, sollten Sie unbedingt eine Verteilungsstatistik anfordern, indem Sie die Klausel WITH DISTRIBUTION für den Befehl **runstats** verwenden. Mit Hilfe der nicht einheitlichen Verteilungsstatistik kann das Optimierungsprogramm zwischen häufig auftretenden und weniger häufig auftretenden Werten unterscheiden.

Operatorname: MSJOIN

Stellt Folgendes dar: Eine Mischverknüpfung, für die die qualifizierten Zeilen aus der äußeren und inneren Tabelle die Reihenfolge des Verknüpfungsvergleichselement aufweisen müssen. Eine Mischverknüpfung wird auch *Mischsuchverknüpfung* oder *Mischverknüpfung nach Sortieren* genannt.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine Mischverknüpfung ist möglich, wenn es ein Verknüpfungsvergleichselement gibt, das die Spalten aus zwei verschiedenen Tabellen gleichsetzt. Sie kann auch aus einer umgeschriebenen Unterabfrage entstehen.

Eine Mischverknüpfung erfordert eine geordnete Eingabe in die zu verknüpfenden Spalten, da die Tabellen in der Regel nur einmal durchsucht werden. Eine geordnete Eingabe erreichen Sie, indem Sie auf einen Index oder auf eine sortierte Tabelle zugreifen.

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Verwenden Sie lokale Vergleichselemente (d. h. Vergleichselemente, die nur auf eine Tabelle verweisen), um die Anzahl der zu verknüpfenden Zeilen zu verringern.
Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

NLJOIN

Operatorname: NLJOIN

Stellt Folgendes dar: Eine verschachtelte Schleifenverknüpfung, die die innere Tabelle (in der Regel über eine Indexsuche) für jede Zeile in der äußeren Tabelle einmal durchsucht.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine verschachtelte Schleifenverknüpfung erfordert kein Verknüpfungsvergleichselement, die Leistung erhöht sich jedoch, wenn ein Verknüpfungsvergleichselement verwendet wird.

Zur Ausführung einer verschachtelten Schleifenverknüpfung haben Sie die folgenden Möglichkeiten:

- Durchsuchen der inneren Tabelle für jede Zeile der äußeren Tabelle, auf die zugegriffen wurde
- Ausführen einer Indexsuchfunktion in der inneren Tabelle für jede Zeile der äußeren Tabelle, auf die zugegriffen wurde

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Eine verschachtelte Schleifenverknüpfung ist eventuell effizienter, wenn es für die Spalten mit Verknüpfungsvergleichselementen in der inneren Tabelle (die Tabelle, die rechts neben dem Operator NLJOIN angezeigt wird) einen Index gibt. Prüfen Sie, ob für die innere Tabelle der Operator TBSCAN anstelle des Operators IXSCAN vorliegt. Ist dies der Fall, sollten Sie einen Index für die verknüpften Spalten hinzufügen.

Eine andere (weniger bedeutende) Möglichkeit, die Verknüpfung effizienter zu machen, ist das Erstellen eines Indexes für die verknüpften Spalten der äußeren Tabelle, so dass die äußere Tabelle geordnet wird.

Weitere Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Zugehörige Informationen:

- Sternverknüpfungen

PIPE

Operatorname: PIPE

Stellt Folgendes dar: Die Übertragung von Zeilen zu anderen Operatoren, ohne dass die Zeilen geändert werden.

(Dieser Operator gilt nur für den Debugmodus.)

RETURN

Operatorname: RETURN

Stellt Folgendes dar: Die Rückgabe von Daten aus einer Abfrage an den Benutzer. Dies ist der letzte Operator im Zugriffsplandiagramm; er zeigt die gesamten Endwerte und den gesamten Aufwand für den Zugriffsplan an.

Dieser Operator stellt eine notwendige Operation dar.

Vorschlag zur Leistungsverbesserung:

- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.

RIDSCN

Operatorname: RIDSCN

Stellt Folgendes dar: Das Durchsuchen einer Liste von Zeilenkennungen (Satz-IDs), die aus mindestens einem Index erhalten wurden.

Dieser Operator wird in folgenden Fällen vom Optimierungsprogramm berücksichtigt:

- Vergleichselemente sind durch OR-Schlüsselworte verbunden, oder es gibt das Vergleichselement IN. Es kann die Technik des logischen Verknüpfens von Indizes über OR (Index ORing) verwendet werden, bei der die Ergebnisse aus mehreren Indexzugriffen auf dieselbe Tabelle kombiniert werden.
- Es ist von Vorteil, einen Vorabesezugriff über Listen für einen einzelnen Indexzugriff zu verwenden, da das Sortieren der Zeilenkennungen vor dem Zugriff auf die Basiszeilen die Eingabe/Ausgabe effizienter macht.

RQUERY

Operatorname: SHIP

Stellt Folgendes dar: Ein Operator, der in dem System zusammengeschlüsselter Datenbanken verwendet wird, um Daten aus einer fernen Datenquelle abzurufen. Dieser Operator wird vom Optimierungsprogramm berücksichtigt, wenn der Operator SHIP die SQL-Anweisung SELECT an eine ferne Datenquelle sendet, um das Abfrageergebnis abzurufen. Die SELECT-Anweisung wird mit der SQL-Programmversion generiert, die von der Datenquelle unterstützt wird, und kann jede beliebige Abfrage enthalten, die von der Datenquelle zugelassen wird.

Vorschlag zur Leistungsverbesserung: Weitere Informationen finden Sie in Kapitel 4 des Handbuchs *Systemverwaltung, Band 2*.

SORT

Operatorname: SORT

Stellt Folgendes dar: Das Sortieren der Zeilen in einer Tabelle in der Reihenfolge mindestens einer Spalte der Tabelle. Optional können doppelte Einträge ausgeschlossen werden.

Ein Sortieren ist erforderlich, wenn kein Index vorhanden ist, der für die geforderte Reihenfolge sorgt, oder wenn ein Sortieren weniger aufwändig ist als eine Indexsuche. Das Sortieren wird in der Regel als letzte Operation ausgeführt, wenn die erforderlichen Zeilen abgerufen wurden, oder um die Daten vor einer Verknüpfung oder Gruppierung zu sortieren.

Wenn die Anzahl der Zeilen hoch ist oder die sortierten Daten nicht über eine Pipe geleitet werden können, erfordert die Operation die aufwändige Generierung von temporären Tabellen.

Weitere Informationen zu Sortierungen finden Sie im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Sie sollten einen Index für die Sortierspalten hinzufügen.
Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.
- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.

- Überprüfen Sie, dass die Größe für den Vorablesezugriff des temporären Systemtabellenbereichs angemessen, d. h. nicht an die Eingabe/Ausgabe gebunden ist. (Wählen Sie **Anweisung->Statistikdaten anzeigen->Tabellenbereiche** aus, um dies zu überprüfen.)
- Wenn häufige große Sortierungen erforderlich sind, sollten Sie die Werte der folgenden Konfigurationsparameter erhöhen:
 - Zwischenspeicher für Sortierlisten (sortheap). Wenn Sie diesen Parameter ändern möchten, klicken Sie die Datenbank in der Steuerzentrale mit Maustaste 2 an, und wählen Sie anschließend **Konfigurieren** aus dem Popup-Menü aus. Wählen Sie die Indexzunge **Leistung** aus dem angezeigten Notizbuch aus.
 - Zwischenspeicher für Sortierlisten (sheapthres). Wenn Sie diesen Parameter ändern möchten, klicken Sie das Datenbankexemplar in der Steuerzentrale mit Maustaste 2 an, und wählen Sie anschließend **Konfigurieren** aus dem Popup-Menü aus. Wählen Sie die Indexzunge **Leistung** aus dem angezeigten Notizbuch aus.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

TBSCAN

Operatorname: TBSCAN

Stellt Folgendes dar: Eine Tabellensuche (Relationsuche), die Zeilen abruft, indem alle erforderlichen Daten direkt aus den Datenseiten gelesen werden.

Dieser Typ des Suchens wird vom Optimierungsprogramm gegenüber einer Indexsuche bevorzugt, wenn Folgendes zutrifft:

- Der durchsuchte Wertebereich kommt häufig vor (d. h. es muss auf den Großteil der Tabelle zugegriffen werden).
- Die Tabelle ist klein.
- Das Indexclustering ist gering.
- Es ist kein Index vorhanden.

Weitere Informationen zur Tabellen- und Indexsuche finden Sie im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Eine Indexsuche ist effizienter als eine Tabellensuche, wenn die Tabelle groß ist und auf die meisten Zeilen der Tabelle nicht zugegriffen wird. Wenn Sie die Wahrscheinlichkeit erhöhen möchten, dass das Optimierungsprogramm in dieser Situation eine Indexsuche verwendet, sollten Sie Indizes für die Spalten hinzufügen, für die es selektive Vergleichselemente gibt.

Weitere Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

- Wenn ein Index bereits vorhanden ist, jedoch nicht verwendet wurde, sollten Sie prüfen, ob es selektive Vergleichselemente in den einzelnen führenden Spalten gibt. Wenn diese Vergleichselemente vorhanden sind, überprüfen Sie als nächstes, dass der Grad der Clusterbildung für den Index hoch ist. (Wenn Sie diese Statistik anzeigen möchten, öffnen Sie das Fenster mit der Tabellenstatistik für die Tabelle unter der Sortierung, und wählen Sie den Druckknopf **Indizes** aus, um das Fenster **Indexstatistik** anzuzeigen.)
- Überprüfen Sie, dass die Größe für den Vorablesezugriff des Tabellenbereichs angemessen, d. h. nicht an die Eingabe/Ausgabe gebunden ist. (Wählen Sie **Anweisung->Statistikdaten anzeigen->Tabellenbereiche** aus, um dies zu überprüfen.)

Weitere Informationen finden Sie im Abschnitt zum Vorablesezugriff auf Daten im Pufferpool im Handbuch *Systemverwaltung*.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Die Statistiken zum Quantil und zu den häufigen Werten bieten Informationen zur Selektivität von Vergleichselementen. Diese Statistikdaten würden beispielsweise dazu verwendet, festzustellen, wann eine Indexsuche gegenüber einer Tabellensuche bevorzugt wird. Verwenden Sie den Befehl **runstats** für eine Tabelle mit der Klausel WITH DISTRIBUTION, um diese Werte zu aktualisieren.

TEMP

Operatorname: TEMP

Stellt Folgendes dar: Das Speichern von Daten in einer temporären Tabelle, damit sie von einem anderen Operator (möglicherweise mehrfach) wieder aus der Tabelle gelesen werden können. Die Tabelle wird spätestens dann entfernt, wenn die SQL-Anweisung verarbeitet wird.

Dieser Operator ist erforderlich, um Unterabfragen auszuwerten oder um Zwischenergebnisse zu speichern. In einigen Fällen (z. B. wenn die Anweisung aktualisiert werden kann) kann dieser Operator obligatorisch sein.

TQUEUE

Operatorname: TQUEUE

Stellt Folgendes dar: Eine Tabellenwarteschlange, die verwendet wird, um Tabellendaten von einem Datenbankagenten zum anderen zu übergeben, wenn mehrere Datenbankagenten eine Abfrage verarbeiten. Im Falle von Parallelität werden mehrere Datenbankagenten verwendet, um eine Abfrage zu verarbeiten.

Tabellenwarteschlangentypen:

- **Lokal:** Die Tabellenwarteschlange wird verwendet, um Daten zwischen Datenbankagenten innerhalb eines Knotens zu übergeben. Eine lokale Tabellenwarteschlange wird für partitionsinterne Parallelität verwendet.
- **Nicht lokal:** Die Tabellenwarteschlange wird verwendet, um Daten zwischen Datenbankagenten auf unterschiedlichen Knoten zu übergeben.

UNION

Operatorname: UNION

Stellt Folgendes dar: Die Verkettung von Datenströmen von Zeilen aus mehreren Tabellen.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu verkettenden Zeilensatz definieren.

UNIQUE

Operatorname: UNIQUE

Stellt Folgendes dar: Das Ausschließen von Zeilen mit doppelten Werten für die angegebenen Spalten.

Vorschlag zur Leistungsverbesserung:

- Dieser Operator ist nicht notwendig, wenn ein eindeutiger Index für die entsprechenden Spalten vorhanden ist.

Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

UPDATE

Operatorname: UPDATE

Stellt Folgendes dar: Das Aktualisieren von Daten in den Zeilen einer Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu aktualisierenden Zeilensatz definieren.

Optimierungsprogramm

Das *Optimierungsprogramm* ist die Komponente des SQL-Compilers, die einen Zugriffsplan für eine SQL-Anweisung in der Datenbearbeitungssprache (DML) auswählt. Dies geschieht, indem der Ausführungsaufwand mehrerer alternativer Zugriffspläne modelliert wird und der Zugriffsplan mit dem geringsten geschätzten Aufwand ausgewählt wird.

Paket

Ein *Paket* ist ein Objekt, das in der Datenbank gespeichert wird und die Informationen enthält, die benötigt werden, um die SQL-Anweisungen zu verarbeiten, die einer Quelldatei eines Anwendungsprogramms zugeordnet sind. Es gibt folgende Möglichkeiten, ein Paket zu generieren:

- Vorkompilieren einer Quelldatei mit dem Befehl **PREP**
- Binden einer Bindedatei, die vom Precompiler mit dem Befehl **BIND** generiert wurde.

Selektivität der Vergleichselemente

Der Begriff *Selektivität* bezieht sich auf die Wahrscheinlichkeit, dass jede beliebige Zeile ein Vergleichselement erfüllt (d. h. wahr ist).

Beispiel: Eine Selektivität von 0,01 (1%) für ein Vergleichselement, das für eine Tabelle mit 1.000.000 Zeilen ausgeführt wird, bedeutet, dass das Vergleichselement schätzungsweise 10.000 Zeilen (1% von 1.000.000) zurückgibt und 990.000 Zeilen löscht.

Ein hoch selektives Vergleichselement (mit einer Selektivität von 0,10 oder weniger) ist sinnvoll. Solche Vergleichselemente geben weniger Zeilen zurück, mit denen zukünftige Operatoren arbeiten müssen. Auf diese Weise wird weniger CPU- und E/A-Zeit benötigt, um die Abfrage zu erfüllen.

Beispiel

Angenommen, Sie haben eine Tabelle mit 1.000.000 Zeilen und die ursprüngliche Abfrage enthält die Klausel ORDER BY, die einen zusätzlichen Schritt zur Sortierung erfordert. Bei einem Vergleichselement mit der Selektivität 0,01 müsste die Sortierung bei etwa 10.000 Zeilen vorgenommen werden. Bei einem weniger selektiven Vergleichselement von 0,50 müsste die Sortierung bei etwa 500.000 Zeilen vorgenommen werden, was mehr CPU- und E/A-Zeit in Anspruch nimmt.

Statisches SQL

Eine *statische SQL*-Anweisung ist in ein Anwendungsprogramm eingebettet. Diese eingebetteten Anweisungen müssen vorkompiliert und in ein *Paket* gebunden werden, bevor die Anwendung ausgeführt werden kann.

Wenn DB2 diese Anweisungen kompiliert, wird für jede Anweisung ein Zugriffsplan erstellt, der auf den Katalogstatistiken und Konfigurationsparametern basiert, die vorlagen, als die Anweisungen vorkompiliert und gebunden wurden.

Diese Zugriffspläne werden immer verwendet, wenn die Anwendung ausgeführt wird; sie ändern sich nicht, bis das Paket neu gebunden wird.

Die Alternative zum statischen SQL ist dynamisches SQL.

Sternverknüpfungen

Eine Reihe von Verknüpfungen wird als Sternverknüpfung betrachtet, wenn eine Fakttable (große zentrale Tabelle) mit mindestens zwei Dimensionstabellen (kleinere Tabellen mit Beschreibungen der Spaltenwerte in der Fakttable) verknüpft wird.

Eine Sternverknüpfung besteht aus drei Hauptteilen:

- Einfache Gleichheitsverknüpfungen
- Logisches Verknüpfen von Indizes über AND (Index ANDing) bei den Ergebnissen der einfachen Gleichheitsverknüpfungen
- Abschließen der einfachen Gleichheitsverknüpfungen

Die Sternverknüpfung stellt zwei oder mehr Verknüpfungen dar, die als Eingabe für einen IXAND-Operator dienen.

Eine einfache Gleichheitsverknüpfung ist eine besondere Form der Verknüpfung, bei der das Ergebnis der Verknüpfung nur die Zeilenkennung (RID) der inneren Tabelle ist, anstelle einer Verknüpfung der Spalten der inneren und äußeren Tabellen.

Sternverknüpfungen verwenden einfache Gleichheitsverknüpfungen, um einem IXAND-Operator Zeilenkennungen zur Verfügung zu stellen. Der IXAND-Operator verstärkt den Filtereffekt der verschiedenen Verknüpfungen. Die Ausgabe des IXAND-Operators wird einem Operator für logisches Verknüpfen von Indizes über OR (Index ORing) zur Verfügung gestellt, der die Zeilenkennungen sortiert und doppelte Zeilen ausschließt, die aus Verknüpfungen resultieren können, die als Eingabe für den IXAND-Operator dienen. Die Zeilen aus der Fakttable werden dann über einen FETCH-Operator abgerufen. Schließlich wird die reduzierte Fakttable mit allen Dimensionstabellen verknüpft, um die Verknüpfungen abzuschließen.

Vorschläge zur Leistungsverbesserung:

- Erstellen Sie für jede der Dimensionstabellenverknüpfung Indizes in der Fakttable.
- Stellen Sie sicher, dass der Schwellenwert für den Sortierspeicher hoch genug ist, um ein Zuordnen des Bitfilters des IXAND-Operators zu ermöglichen. Für Sternverknüpfungen kann dies bis zu 12 MB oder 3.000 4KB-Seiten erfordern. Bei partitionsinterner Parallelität wird der Bitfilter von demselben gemeinsam benutzten Speichersegment zugeordnet wie der Zwischenspeicher der Datenbank, er wird jedoch von dem Zwischenspeicher für Sortierlisten (und vom Schwellenwert für den Sortierspeicher im Exemplar) begrenzt. Der gemeinsam benutzte Speicher wird daher von dem Zwischenspeicher für Sortierlisten und dem Schwellenwert für den Sortierspeicher gesteuert und kann unter Umständen mehr als 12 MB erfordern.

- Wenden Sie filternde Vergleichselemente auf die Dimensionstabellen an. Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren..

Tabellenbereich

Es ist leichter, sehr große Datenbanken zu verwalten, wenn Sie sie in separat verwaltete Teile, so genannte *Tabellenbereiche*, partitionieren.

Mit Hilfe eines Tabellenbereichs können Sie die Speicherposition von Daten bestimmten logischen Einheiten oder Abschnitten von Einheiten zuordnen. Wenn Sie beispielsweise eine Tabelle erstellen, können Sie angeben, dass die Indizes oder Spalten für große Objekte (LOBs) von den übrigen Tabellendaten getrennt werden.

Ein Tabellenbereich kann über mindestens eine physische Speichereinheit (Behälter) verteilt sein, um die Leistung zu erhöhen. Es wird jedoch empfohlen, dass alle Einheiten oder Behälter innerhalb des Tabellenbereichs ähnliche Leistungskenndaten haben.

Ein Tabellenbereich kann auf zwei verschiedene Weise verwaltet werden: als SMS- oder als DMS-Tabellenbereich.

Vergleichselement

Ein *Vergleichselement* ist ein Element einer Suchbedingung, das eine Vergleichsoperation ausdrückt oder angibt. Vergleichselemente sind in Klauseln enthalten, die mit WHERE oder HAVING beginnen.

Beispiel: In der folgenden SQL-Anweisung:

```
SELECT * FROM SAMPLE WHERE NAME = 'SMITH' AND DEPT = 895 AND YEARS > 5
```

lauten die Vergleichselemente wie folgt: NAME = 'SMITH'; DEPT = 895; und YEARS > 5.

Vergleichselemente fallen in eine der folgenden Kategorien von sehr effizient bis weniger effizient:

1. Start/Stop-Bedingungen schränken eine Indexsuche ein. (Diese Bedingungen werden auch bereichsbegrenzende Vergleichselemente genannt.)
2. Indexseitenvergleichselemente (im Index als Suchargument verwendbare Vergleichselemente) können aus einem Index ausgewertet werden, da die Spalten, die zum Vergleichselement gehören, Teil des Indexschlüssels sind.

3. Datenseitenvergleichselemente (in Daten als Suchargument verwendbare Vergleichselemente) können aus einem Index nicht ausgewertet werden, können jedoch ausgewertet werden, während die Zeilen im Puffer bleiben.
4. Verbleibende Vergleichselemente erfordern in der Regel E/A über den einfachen Zugriff auf eine Basistabelle hinaus, und müssen angewendet werden, nachdem die Daten aus der Pufferseite kopiert wurden. Sie umfassen Vergleichselemente, die Unterabfragen enthalten oder die LONG VARCHAR oder LOB-Daten lesen, die in Dateien getrennt von der Tabelle gespeichert werden.

Wenn Sie Vergleichselemente entwerfen, sollten Sie versuchen, die höchstmögliche Selektivität zu erreichen, so dass so wenig Zeilen wie möglich zurückgegeben werden.

Die folgenden Vergleichselementtypen sind die effektivsten und am häufigsten verwendeten Vergleichselemente:

- Ein *einfaches Gleichheitsverknüpfungselement* ist für eine Mischverknüpfung erforderlich. Es hat die Form `tabelle1.spalte = tabelle2.spalte` und ermöglicht, dass Spalten in zwei verschiedenen Tabellen gleichgesetzt werden, so dass die Tabellen verknüpft werden können.
- Ein *lokales Vergleichselement* wird nur auf eine Tabelle angewendet.

Weitere Informationen finden Sie in den Abschnitten zu den Datenzugriffskonzepten und zur Optimierung im Handbuch *Systemverwaltung*.

Visual Explain

Anmerkung: Ab Version 6 kann Visual Explain nicht mehr über die Befehlszeile aufgerufen werden. Es kann jedoch weiterhin über verschiedene Datenbankobjekte in der Steuerzentrale aufgerufen werden. In dieser Version wird in der Dokumentation weiterhin der Name Visual Explain verwendet.

Mit Visual Explain können Sie den Zugriffsplan für mit EXPLAIN bearbeitete SQL-Anweisungen als Diagramm anzeigen. Sie können die Informationen aus dem Diagramm verwenden, um Ihre SQL-Abfragen so zu optimieren, dass die Leistung erhöht wird.

In einem Zugriffsplandiagramm werden folgende Details angezeigt:

- Tabellen (und ihre zugeordneten Spalten) und Indizes
- Operatoren (wie Tabellensuche, Sortierungen und Verknüpfungen)
- Tabellenbereiche und Funktionen

Sie können Visual Explain auch für Folgendes verwenden:

- Anzeigen der Statistikdaten, die bei der Optimierung verwendet wurden. Sie können diese Statistikdaten dann mit den aktuellen Katalogstatistiken vergleichen, um festzustellen, ob ein erneutes Binden des Pakets die Leistung erhöhen kann.
- Feststellen, ob ein Index für den Zugriff auf eine Tabelle verwendet wurde. Wenn kein Index verwendet wurde, kann Visual Explain helfen, festzustellen, welche Spalten von einer Indexierung profitieren würden.
- Anzeigen der Auswirkungen von Ausführungen verschiedener Optimierungstechniken, indem die vorherige Version mit der resultierenden Version des Zugriffsplandiagramms für eine Abfrage verglichen wird.
- Erhalten von Informationen zu jeder Operation im Zugriffsplan, einschließlich des gesamten geschätzten Aufwands und der Anzahl der abgerufenen Zeilen (Kardinalität).

Vom Betriebssystem verwaltete Tabellenbereiche (SMS)

Es gibt zwei Typen von Tabellenbereichen, die in einer Datenbank vorhanden sein können: vom Betriebssystem verwaltete Bereiche (SMS) und vom Datenbankmanager verwaltete Bereiche (DMS). Ein SMS-Tabellenbereich wird vom Betriebssystem verwaltet, das die Datenbankdaten in einem Bereich speichert, der zugeordnet wird, wenn ein Tabellenbereich erstellt wird. Die Tabellenbereichsdefinition enthält eine Liste mit mindestens einem Verzeichnispfad, in dem diese Daten gespeichert werden.

Das Dateisystem verwaltet die Zuordnung und Verwaltung des Datenträgerspeichers.

SMS- und DMS-Tabellenbereiche können in derselben Datenbank nebeneinander existieren.

Vom Datenbankmanager verwalteter Tabellenbereich (DMS)

Es gibt zwei Typen von Tabellenbereichen, die in einer Datenbank vorhanden sein können: vom Datenbankmanager verwaltete Bereiche (DMS) und vom Betriebssystem verwaltete Bereiche (SMS).

DMS-Tabellenbereiche werden vom Datenbankmanager verwaltet. Sie werden so entwickelt und optimiert, dass sie seinen Anforderungen entsprechen.

Die Definition des DMS-Tabellenbereichs enthält eine Liste mit Dateien (oder Einheiten), in die die Datenbankdaten in ihrem DMS-Tabellenbereichsformat gespeichert werden.

Sie können einem vorhandenen DMS-Tabellenbereich zuvor zugeordnete Dateien (oder Einheiten) hinzufügen, um die Speicherkapazität zu erhöhen. Der Datenbankmanager gleicht die vorhandenen Daten in allen Behältern, die zu diesem Tabellenbereich gehören, automatisch neu aus.

DMS- und SMS-Tabellenbereiche können in derselben Datenbank nebeneinander existieren.

Zugriffsplan

Es werden bestimmte Daten benötigt, um eine mit EXPLAIN bearbeitbare SQL-Anweisung aufzulösen. Ein *Zugriffsplan* gibt die Reihenfolge der Operationen beim Zugriff auf diese Daten an. Es werden nicht nur Statistikdaten für ausgewählte Tabellen, Indizes oder Spalten angezeigt, sondern auch Merkmale für Operatoren, globale Informationen, wie Statistikdaten zu Tabellenbereichen und Funktionen, sowie Konfigurationsparameter, die für die Optimierung relevant sind. Mit Visual Explain können Sie den Zugriffsplan für eine SQL-Anweisung in grafischer Form anzeigen.

Das Optimierungsprogramm produziert einen Zugriffsplan, wenn eine mit EXPLAIN bearbeitbare SQL-Anweisung kompiliert wird. Dies geschieht für statische Anweisungen während der Vorbereitungs-/Bindezeit und für dynamische Anweisungen während der Bearbeitungszeit.

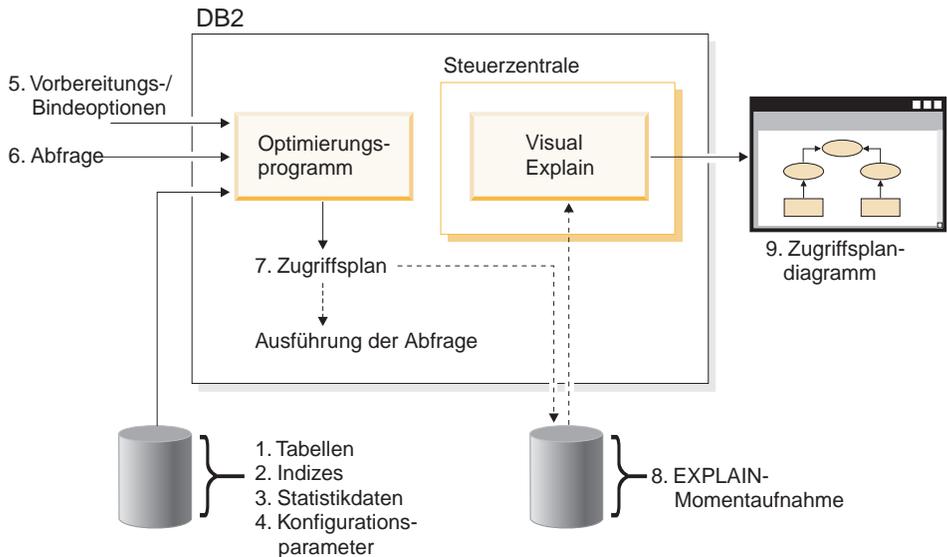
Sie sollten sich darüber im Klaren sein, dass es sich bei einem Zugriffsplan um eine *Schätzung* handelt, die auf den verfügbaren Informationen basiert. Das Optimierungsprogramm basiert seine Schätzungen auf den folgenden Informationen:

- Statistikdaten in Systemkatalogtabellen (wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.)
- Konfigurationsparameter
- Bindeoptionen
- Die Abfrageoptimierungsklasse

Die Informationen zum Aufwand, die mit einem Zugriffsplan verbunden sind, sind die *beste Schätzung* der Ressourcennutzung für eine Abfrage durch das Optimierungsprogramm. Die tatsächlich abgelaufene Zeit für eine Abfrage kann abhängig von Faktoren außerhalb des DB2-Geltungsbereichs (z. B. die Anzahl anderer Anwendungen, die zur selben Zeit ausgeführt werden) variieren. Die tatsächlich abgelaufene Zeit kann während der Ausführung der Abfrage mit Hilfe des Performance Monitors gemessen werden.

Zugriffsplandiagramm

Visual Explain verwendet Informationen aus einer Vielzahl von Quellen, um ein Zugriffsplandiagramm zu produzieren (siehe Abbildung weiter unten). Auf der Basis verschiedener Eingaben wählt das Optimierungsprogramm einen Zugriffsplan aus, und Visual Explain zeigt ihn in einem *Zugriffsplandiagramm* an. Die Knoten im Diagramm stellen Tabellen und Indizes sowie alle dafür ausgeführten Operationen dar. Die Verbindungen zwischen den Knoten stellen den Datenfluss dar.



Die folgende Liste mit Tasks entspricht den Tasks in der oben aufgeführten Abbildung. (Gestrichelte Linien geben die Schritte an, die für Visual Explain erforderlich sind.)

1. Optimieren Sie Ihren Tabellenentwurf, und reorganisieren Sie die Tabellendaten.
2. Erstellen Sie die entsprechenden Indizes.
3. Verwenden Sie den Befehl **runstats**, um dem Optimierungsprogramm die aktuellen Statistikdaten zur Verfügung zu stellen.
4. Wählen Sie die entsprechenden Konfigurationsparameter aus.
5. Wählen Sie die entsprechenden Bindeoptionen aus.
6. Entwerfen Sie Abfragen, mit denen nur erforderliche Daten abgerufen werden.
7. Erstellen Sie einen Zugriffsplan.
8. Erstellen Sie EXPLAIN-Momentaufnahmen.
9. Zeigen Sie ein Zugriffsplandiagramm an, und verwenden Sie es.

Beispiel: Wenn Sie Visual Explain verwenden möchten, aktualisieren Sie zunächst die aktuellen Statistikdaten mit Hilfe des Befehls **runstats** in den Tabellen und Indizes, die von der Anweisung verwendet werden. Diese Statistikdaten, die Konfigurationsparameter, die Bindeoptionen sowie die Abfrage selbst werden vom Optimierungsprogramm verwendet, um einen Zugriffsplan und eine EXPLAIN-Momentaufnahme zu erstellen, wenn das Paket gebunden wird. Visual Explain verwendet die resultierende EXPLAIN-Momentaufnahme, um das Zugriffsplandiagramm für die Anweisung anzuzeigen.

Zugriffsplandiagrammknoten

Das Zugriffsplandiagramm besteht aus einer Baumstruktur, die die *Knoten* anzeigt. Diese Knoten stellen Folgendes dar:

- Tabellen, als Rechtecke dargestellt
- Indizes, als Rauten dargestellt
- Operatoren, als Achtecke (8 Seiten) dargestellt. TQUEUE-Operatoren, als Parallelogramme dargestellt
- Tabellenfunktionen, als Sechsecke (6 Seiten) dargestellt

Anhang B. Alphabetische Liste der Visual Explain-Operatoren

CMPEXP

Operatorname: CMPEXP

Stellt Folgendes dar: Die Verarbeitung von Ausdrücken, die für temporäre oder endgültige Ergebnisse erforderlich sind.

(Dieser Operator gilt nur für den Debugmodus.)

DELETE

Operatorname: DELETE

Stellt Folgendes dar: Das Löschen von Zeilen aus einer Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu löschenden Zeilensatz definieren.

Vorschlag zur Leistungsverbesserung:

- Wenn Sie alle Zeilen aus einer Tabelle löschen, sollten Sie die Anweisung DROP TABLE oder den Befehl **LOAD REPLACE** verwenden.

EISCAN

Operatorname: EISCAN

Stellt Folgendes dar: Dieser Operator durchsucht einen benutzerdefinierten Index, um einen verringerten Datenstrom von Zeilen zu produzieren.

Beim Durchsuchen werden die mehrfachen Start/Stop-Bedingungen aus der Bereichsgenerierungsfunktion verwendet, die vom Benutzer eingerichtet wurde.

Diese Operation wird ausgeführt, um den Satz qualifizierender Zeilen einzuschränken, bevor auf die Basistabelle (basierend auf Vergleichselementen) zugegriffen wird.

Vorschlag zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Indexes korrigiert werden.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

FETCH

Operatorname: FETCH

Stellt Folgendes dar: Das Abrufen von Spalten aus einer Tabelle mit Hilfe einer speziellen Zeilenkennung (RID).

Vorschläge zur Leistungsverbesserung:

- Erweitern Sie die Indexschlüssel, um die abgerufenen Spalten aufzunehmen, so dass auf die Datenseiten nicht zugegriffen werden muss.
- Suchen Sie den Index, der mit dem Abruf verbunden ist, und klicken Sie seinen Knoten doppelt an, um das Fenster mit den Statistikdaten anzuzeigen. Stellen Sie sicher, dass der Grad der Clusterbildung für den Index hoch ist.
- Erhöhen Sie die Puffergröße, wenn die Eingabe/Ausgabe (E/A) bei dem Abruf größer als die Anzahl der Seiten in der Tabelle ist.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Die Statistiken zum Quantil und zu den häufigen Werten bieten Informationen zur Selektivität von Vergleichselementen, die bestimmen, wann eine Indexsuche gegenüber einer Tabellensuche bevorzugt wird. Verwenden Sie den Befehl **runstats** für eine Tabelle mit der Klausel WITH DISTRIBUTION, um diese Statistikdaten zu aktualisieren.

FILTER

Operatorname: FILTER

Stellt Folgendes dar: Die Anwendung von verbleibenden Vergleichselementen, so dass Daten auf der Basis der Bedingungen der Vergleichselemente gefiltert werden.

Vorschläge zur Leistungsverbesserung:

- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.
- Stellen Sie sicher, dass die Optimierungsklasse mindestens 3 beträgt, so dass das Optimierungsprogramm eine Verknüpfung anstelle einer Unterabfrage verwendet. Wenn dies nicht möglich ist, sollten Sie versuchen, die SQL-Abfrage selbst umzuschreiben, um die Unterabfrage auszuschließen. Ein Beispiel finden Sie im Abschnitt zum Umschreiben von Abfragen durch den SQL-Compiler im Handbuch *Systemverwaltung*.

GENROW

Operatorname: GENROW

Stellt Folgendes dar: Eine integrierte Funktion, die eine Tabelle mit Zeilen generiert, wobei keine Eingaben aus Tabellen, Indizes oder Operatoren verwendet werden.

GENROW kann vom Optimierungsprogramm verwendet werden, um Datenzeilen zu generieren (z. B. für die Anweisung INSERT oder für einige Listen IN, die in Verknüpfungen umgesetzt werden).

Wenn Sie die geschätzten Statistikdaten für die von der Funktion GENROW generierten Tabellen anzeigen möchten, klicken Sie den Knoten dieser Funktion doppelt an.

GRPBY

Operatorname: GRPBY

Stellt Folgendes dar: Das Gruppieren von Zeilen nach gemeinsamen Werten zugewiesener Spalten oder Funktionen. Diese Operation ist erforderlich, um eine Gruppe von Werten herzustellen oder um festgelegte Funktionen auszuwerten.

Wenn keine GROUP BY-Spalten angegeben werden, kann der Operator GRPBY dennoch verwendet werden, wenn es Spaltenfunktionen in der SELECT-Liste gibt, die angeben, dass die gesamte Tabelle als einzelne Gruppe behandelt werden soll, wenn die Spaltenberechnung vorgenommen wird.

Vorschläge zur Leistungsverbesserung:

- Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsplanaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu gruppierenden Zeilensatz definieren.
- Gehen Sie wie folgt vor, um die Leistung der Anweisung SELECT zu verbessern, die eine Spaltenfunktion, jedoch keine GROUP BY-Klausel enthält:
 - Erstellen Sie für die Spaltenfunktion MIN(C) einen aufsteigenden Index für C.
 - Erstellen Sie für die Spaltenfunktion MAX(C) einen absteigenden Index für C.

HSJOIN

Operatorname: HSJOIN

Stellt Folgendes dar: Eine Hashverknüpfung, für die die qualifizierten Zeilen aus Tabellen im Hashverfahren verknüpft werden, um eine direkte Verknüpfung zu ermöglichen, ohne dass der Inhalt der Tabellen zuvor geordnet wird.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine Hashverknüpfung ist möglich, wenn es ein Verknüpfungsvergleichselement gibt, das die Spalten aus zwei verschiedenen Tabellen gleichsetzt. Die Verknüpfungsvergleichselemente müssen genau denselben Datentyp aufweisen. Hashverknüpfungen können auch aus umgeschriebenen Unterabfragen entstehen, wie dies bei NLJOIN der Fall ist.

Bei einer Hashverknüpfung müssen die Eingabetabellen nicht geordnet sein. Die Verknüpfung wird vorgenommen, indem die innere Tabelle der Hashverknüpfung durchsucht wird und eine Referenztable generiert wird, indem die zu verknüpfenden Spaltenwerte im Hashverfahren verknüpft werden. Anschließend wird die äußere Tabelle gelesen, wobei die zu verknüpfenden Spaltenwerte im Hashverfahren verknüpft werden und in der Referenztable, die für die innere Tabelle generiert wurde, überprüft werden.

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Verwenden Sie lokale Vergleichselemente (d. h. Vergleichselemente, die nur auf eine Tabelle verweisen), um die Anzahl der zu verknüpfenden Zeilen zu verringern.
- Erhöhen Sie die Größe des Zwischenspeichers für Sortierlisten, damit die Hashreferenztable im Speicher gehalten werden kann.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

INSERT

Operatorname: INSERT

Stellt Folgendes dar: Das Einfügen von Zeilen in eine Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffspaufwand verbessern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den einzufügenden Zeilensatz definieren.

IXAND

Operatorname: IXAND

Stellt Folgendes dar: Das logische Verknüpfen der Ergebnisse von Mehrfachindexsuchen mit AND und Techniken für die dynamische Bitzuordnung. Der Operator ermöglicht es, dass Vergleichselemente, die über AND verknüpft sind, auf mehrere Indizes angewendet werden, um die Zugriffe auf zugrundeliegende Tabellen auf ein Minimum zu reduzieren.

Dieser Operator wird aus folgenden Gründen ausgeführt:

- Einschränken des Zeilensatzes, bevor auf die Basistabelle zugegriffen wird
- Verknüpfen von Vergleichselementen über AND, die auf mehrere Indizes angewendet werden
- Verknüpfen von Ergebnissen aus einfachen Gleichheitsverknüpfungen über AND, die in Sternverknüpfungen verwendet werden

Vorschläge zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Indexes korrigiert werden.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.
- Im Allgemeinen ist eine Indexsuche am effektivsten, wenn sich nur wenige Zeilen qualifizieren. Das Optimierungsprogramm verwendet die Statistikdaten, die für die in den Vergleichselementen verwiesenen Spalten verfügbar sind, um die Anzahl der sich qualifizierenden Zeilen zu schätzen. Wenn einige Werte häufiger auftreten als andere, sollten Sie unbedingt eine Verteilungstatistik anfordern, indem Sie die Klausel WITH DISTRIBUTION für den Befehl **runstats** verwenden. Mit Hilfe der nicht einheitlichen

Verteilungsstatistik kann das Optimierungsprogramm zwischen häufig auftretenden und weniger häufig auftretenden Werten unterscheiden.

- IXAND kann am besten für einspaltige Indizes genutzt werden, da Start- und Stoppschlüssel bei der Verwendung von IXAND kritisch sind.
- Für Sternverknüpfungen erstellen Sie einspaltige Indizes für die Spalten mit der höchsten Selektivität in der Faktabelle und den dazugehörigen Dimensionstabellen.

IXSCAN

Operatorname: IXSCAN

Stellt Folgendes dar: Das Durchsuchen eines Indexes, um einen verringerten Datenstrom von Zeilen zu produzieren. Beim Durchsuchen können optionale Start/Stop-Bedingungen verwendet werden oder indexierbare Vergleichselemente angewendet werden, die auf Spalten des Indexes verweisen.

Diese Operation wird ausgeführt, um den Satz qualifizierender Zeilen einzuschränken, bevor auf die Basistabelle (basierend auf Vergleichselementen) zugegriffen wird.

Weitere Informationen finden Sie im Abschnitt zur Indexsuche im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Im Laufe der Zeit wird ein Index durch die Datenbankaktualisierungen möglicherweise fragmentiert, so dass mehr Indexseiten als erforderlich benötigt werden. Dies kann durch Löschen und erneutes Erstellen oder durch Reorganisieren des Index korrigiert werden.
- Wenn auf mindestens zwei Tabellen zugegriffen wird, kann ein Zugriff auf die innere Tabelle über einen Index effektiver werden, wenn ein Index in der verknüpften Spalte der äußeren Tabelle eingerichtet wird.

Weitere Richtlinien zu Indizes finden Sie in der Onlinehilfefunktion für Visual Explain.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.
- Im Allgemeinen ist eine Indexsuche am effektivsten, wenn sich nur wenige Zeilen qualifizieren. Das Optimierungsprogramm verwendet die Statistikdaten, die für die in den Vergleichselementen verwiesenen Spalten verfügbar sind, um die Anzahl der sich qualifizierenden Zeilen zu schätzen. Wenn einige Werte häufiger auftreten als andere, sollten Sie unbedingt eine Verteilungsstatistik anfordern, indem Sie die Klausel WITH DISTRIBUTION für den Befehl **runstats** verwenden. Mit Hilfe der nicht einheitlichen Verteilungsstatistik kann das Optimierungsprogramm zwischen häufig auftretenden und weniger häufig auftretenden Werten unterscheiden.

MSJOIN

Operatorname: MSJOIN

Stellt Folgendes dar: Eine Mischverknüpfung, für die die qualifizierten Zeilen aus der äußeren und inneren Tabelle die Reihenfolge des Verknüpfungsvergleichselement aufweisen müssen. Eine Mischverknüpfung wird auch *Mischsuchverknüpfung* oder *Mischverknüpfung nach Sortieren* genannt.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine Mischverknüpfung ist möglich, wenn es ein Verknüpfungsvergleichselement gibt, das die Spalten aus zwei verschiedenen Tabellen gleichsetzt. Sie kann auch aus einer umgeschriebenen Unterabfrage entstehen.

Eine Mischverknüpfung erfordert eine geordnete Eingabe in die zu verknüpfenden Spalten, da die Tabellen in der Regel nur einmal durchsucht werden. Eine geordnete Eingabe erreichen Sie, indem Sie auf einen Index oder auf eine sortierte Tabelle zugreifen.

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Verwenden Sie lokale Vergleichselemente (d. h. Vergleichselemente, die nur auf eine Tabelle verweisen), um die Anzahl der zu verknüpfenden Zeilen zu verringern.
Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

NLJOIN

Operatorname: NLJOIN

Stellt Folgendes dar: Eine verschachtelte Schleifenverknüpfung, die die innere Tabelle (in der Regel über eine Indexsuche) für jede Zeile in der äußeren Tabelle einmal durchsucht.

Eine Verknüpfung ist erforderlich, wenn in einer FROM-Klausel auf mehrere Tabellen verwiesen wird. Eine verschachtelte Schleifenverknüpfung erfordert kein Verknüpfungsvergleichselement, die Leistung erhöht sich jedoch, wenn ein Verknüpfungsvergleichselement verwendet wird.

Zur Ausführung einer verschachtelten Schleifenverknüpfung haben Sie die folgenden Möglichkeiten:

- Durchsuchen der inneren Tabelle für jede Zeile der äußeren Tabelle, auf die zugegriffen wurde
- Ausführen einer Indexsuchfunktion in der inneren Tabelle für jede Zeile der äußeren Tabelle, auf die zugegriffen wurde

Weitere Informationen finden Sie im Abschnitt zu Verknüpfungskonzepten im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Eine verschachtelte Schleifenverknüpfung ist eventuell effizienter, wenn es für die Spalten mit Verknüpfungsvergleichselementen in der inneren Tabelle (die Tabelle, die rechts neben dem Operator NLJOIN angezeigt wird) einen Index gibt. Prüfen Sie, ob für die innere Tabelle der Operator TBSCAN anstelle des Operators IXSCAN vorliegt. Ist dies der Fall, sollten Sie einen Index für die verknüpften Spalten hinzufügen.

Eine andere (weniger bedeutende) Möglichkeit, die Verknüpfung effizienter zu machen, ist das Erstellen eines Indexes für die verknüpften Spalten der äußeren Tabelle, so dass die äußere Tabelle geordnet wird.

Weitere Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Zugehörige Informationen:

- Sternverknüpfungen

PIPE

Operatorname: PIPE

Stellt Folgendes dar: Die Übertragung von Zeilen zu anderen Operatoren, ohne dass die Zeilen geändert werden.

(Dieser Operator gilt nur für den Debugmodus.)

RETURN

Operatorname: RETURN

Stellt Folgendes dar: Die Rückgabe von Daten aus einer Abfrage an den Benutzer. Dies ist der letzte Operator im Zugriffsplandiagramm; er zeigt die gesamten Endwerte und den gesamten Aufwand für den Zugriffsplan an.

Dieser Operator stellt eine notwendige Operation dar.

Vorschlag zur Leistungsverbesserung:

- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.

RIDSCN

Operatorname: RIDSCN

Stellt Folgendes dar: Das Durchsuchen einer Liste von Zeilenkennungen (Satz-IDs), die aus mindestens einem Index erhalten wurden.

Dieser Operator wird in folgenden Fällen vom Optimierungsprogramm berücksichtigt:

- Vergleichselemente sind durch OR-Schlüsselworte verbunden, oder es gibt das Vergleichselement IN. Es kann die Technik des logischen Verknüpfens von Indizes über OR (Index ORing) verwendet werden, bei der die Ergebnisse aus mehreren Indexzugriffen auf dieselbe Tabelle kombiniert werden.
- Es ist von Vorteil, einen Vorabselektzugriff über Listen für einen einzelnen Indexzugriff zu verwenden, da das Sortieren der Zeilenkennungen vor dem Zugriff auf die Basiszeilen die Eingabe/Ausgabe effizienter macht.

RQUERY

Operatorname: SHIP

Stellt Folgendes dar: Ein Operator, der in dem System zusammengeschlusener Datenbanken verwendet wird, um Daten aus einer fernen Datenquelle abzurufen. Dieser Operator wird vom Optimierungsprogramm berücksichtigt, wenn der Operator SHIP die SQL-Anweisung SELECT an eine ferne Datenquelle sendet, um das Abfrageergebnis abzurufen. Die SELECT-Anweisung wird mit der SQL-Programmversion generiert, die von der Datenquelle unterstützt wird, und kann jede beliebige Abfrage enthalten, die von der Datenquelle zugelassen wird.

Vorschlag zur Leistungsverbesserung: Weitere Informationen finden Sie in Kapitel 4 des Handbuchs *Systemverwaltung*, Band 2.

SORT

Operatorname: SORT

Stellt Folgendes dar: Das Sortieren der Zeilen in einer Tabelle in der Reihenfolge mindestens einer Spalte der Tabelle. Optional können doppelte Einträge ausgeschlossen werden.

Ein Sortieren ist erforderlich, wenn kein Index vorhanden ist, der für die geforderte Reihenfolge sorgt, oder wenn ein Sortieren weniger aufwändig ist als eine Indexsuche. Das Sortieren wird in der Regel als letzte Operation ausgeführt, wenn die erforderlichen Zeilen abgerufen wurden, oder um die Daten vor einer Verknüpfung oder Gruppierung zu sortieren.

Wenn die Anzahl der Zeilen hoch ist oder die sortierten Daten nicht über eine Pipe geleitet werden können, erfordert die Operation die aufwändige Generierung von temporären Tabellen.

Weitere Informationen zu Sortierungen finden Sie im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Sie sollten einen Index für die Sortierspalten hinzufügen.
Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.
- Stellen Sie sicher, dass Sie Vergleichselemente verwendet haben, die nur die erforderlichen Daten abrufen. Stellen Sie beispielsweise sicher, dass der Selektivitätswert für die Vergleichselemente den Teil der Tabelle darstellt, der zurückgegeben werden soll.
- Überprüfen Sie, dass die Größe für den Vorabesezugriff des temporären Systemtabellenbereichs angemessen, d. h. nicht an die Eingabe/Ausgabe gebunden ist. (Wählen Sie **Anweisung->Statistikdaten anzeigen->Tabellenbereiche** aus, um dies zu überprüfen.)
- Wenn häufige große Sortierungen erforderlich sind, sollten Sie die Werte der folgenden Konfigurationsparameter erhöhen:
 - Zwischenspeicher für Sortierlisten (sortheap). Wenn Sie diesen Parameter ändern möchten, klicken Sie die Datenbank in der Steuerzentrale mit Maustaste 2 an, und wählen Sie anschließend **Konfigurieren** aus dem Popup-Menü aus. Wählen Sie die Indexzunge **Leistung** aus dem angezeigten Notizbuch aus.
 - Zwischenspeicher für Sortierlisten (sheapthres). Wenn Sie diesen Parameter ändern möchten, klicken Sie das Datenbankexemplar in der Steuerzentrale mit Maustaste 2 an, und wählen Sie anschließend **Konfigurieren** aus dem Popup-Menü aus. Wählen Sie die Indexzunge **Leistung** aus dem angezeigten Notizbuch aus.
- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Operatormame: TBSCAN

Stellt Folgendes dar: Eine Tabellensuche (Relationssuche), die Zeilen abrufft, indem alle erforderlichen Daten direkt aus den Datensichten gelesen werden.

Dieser Typ des Suchens wird vom Optimierungsprogramm gegenüber einer Indexsuche bevorzugt, wenn Folgendes zutrifft:

- Der durchsuchte Wertebereich kommt häufig vor (d. h. es muss auf den Großteil der Tabelle zugegriffen werden).
- Die Tabelle ist klein.
- Das Indexclustering ist gering.
- Es ist kein Index vorhanden.

Weitere Informationen zur Tabellen- und Indexsuche finden Sie im Handbuch *Systemverwaltung*.

Vorschläge zur Leistungsverbesserung:

- Eine Indexsuche ist effizienter als eine Tabellensuche, wenn die Tabelle groß ist und auf die meisten Zeilen der Tabelle nicht zugegriffen wird. Wenn Sie die Wahrscheinlichkeit erhöhen möchten, dass das Optimierungsprogramm in dieser Situation eine Indexsuche verwendet, sollten Sie Indizes für die Spalten hinzufügen, für die es selektive Vergleichselemente gibt.

Weitere Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

- Wenn ein Index bereits vorhanden ist, jedoch nicht verwendet wurde, sollten Sie prüfen, ob es selektive Vergleichselemente in den einzelnen führenden Spalten gibt. Wenn diese Vergleichselemente vorhanden sind, überprüfen Sie als nächstes, dass der Grad der Clusterbildung für den Index hoch ist. (Wenn Sie diese Statistik anzeigen möchten, öffnen Sie das Fenster mit der Tabellenstatistik für die Tabelle unter der Sortierung, und wählen Sie den Druckknopf **Indizes** aus, um das Fenster **Indexstatistik** anzuzeigen.)
- Überprüfen Sie, dass die Größe für den Vorablesezugriff des Tabellenbereichs angemessen, d. h. nicht an die Eingabe/Ausgabe gebunden ist. (Wählen Sie **Anweisung->Statistikdaten anzeigen->Tabellenbereiche** aus, um dies zu überprüfen.)

Weitere Informationen finden Sie im Abschnitt zum Vorablesezugriff auf Daten im Pufferpool im Handbuch *Systemverwaltung*.

- Wenn die Statistikdaten nicht aktuell sind, sollten Sie sie mit dem Befehl **runstats** aktualisieren.

Die Statistiken zum Quantil und zu den häufigen Werten bieten Informationen zur Selektivität von Vergleichselementen. Diese Statistikdaten würden beispielsweise dazu verwendet, festzustellen, wann eine Indexsuche gegen-

über einer Tabellensuche bevorzugt wird. Verwenden Sie den Befehl **runstats** für eine Tabelle mit der Klausel WITH DISTRIBUTION, um diese Werte zu aktualisieren.

TEMP

Operatorname: TEMP

Stellt Folgendes dar: Das Speichern von Daten in einer temporären Tabelle, damit sie von einem anderen Operator (möglicherweise mehrfach) wieder aus der Tabelle gelesen werden können. Die Tabelle wird spätestens dann entfernt, wenn die SQL-Anweisung verarbeitet wird.

Dieser Operator ist erforderlich, um Unterabfragen auszuwerten oder um Zwischenergebnisse zu speichern. In einigen Fällen (z. B. wenn die Anweisung aktualisiert werden kann) kann dieser Operator obligatorisch sein.

TQUEUE

Operatorname: TQUEUE

Stellt Folgendes dar: Eine Tabellenwarteschlange, die verwendet wird, um Tabellendaten von einem Datenbankagenten zum anderen zu übergeben, wenn mehrere Datenbankagenten eine Abfrage verarbeiten. Im Falle von Parallelität werden mehrere Datenbankagenten verwendet, um eine Abfrage zu verarbeiten.

Tabellenwarteschlangentypen:

- **Lokal:** Die Tabellenwarteschlange wird verwendet, um Daten zwischen Datenbankagenten innerhalb eines Knotens zu übergeben. Eine lokale Tabellenwarteschlange wird für partitionsinterne Parallelität verwendet.
 - **Nicht lokal:** Die Tabellenwarteschlange wird verwendet, um Daten zwischen Datenbankagenten auf unterschiedlichen Knoten zu übergeben.
-

UNION

Operatorname: UNION

Stellt Folgendes dar: Die Verkettung von Datenströmen von Zeilen aus mehreren Tabellen.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu verkettenden Zeilensatz definieren.

UNIQUE

Operatorname: UNIQUE

Stellt Folgendes dar: Das Ausschließen von Zeilen mit doppelten Werten für die angegebenen Spalten.

Vorschlag zur Leistungsverbesserung:

- Dieser Operator ist nicht notwendig, wenn ein eindeutiger Index für die entsprechenden Spalten vorhanden ist.
Richtlinien zu Indizes finden Sie im Abschnitt zum Erstellen von entsprechenden Indizes in der Onlinehilfefunktion für Visual Explain.

UPDATE

Operatorname: UPDATE

Stellt Folgendes dar: Das Aktualisieren von Daten in den Zeilen einer Tabelle.

Dieser Operator stellt eine notwendige Operation dar. Wenn Sie den Zugriffsaufwand verringern möchten, sollten Sie sich auf andere Operatoren (wie Suchen oder Verknüpfungen) konzentrieren, die den zu aktualisierenden Zeilensatz definieren.

Anhang C. DB2-Konzepte

Datenbanken

Eine relationale Datenbank stellt Daten in Form von Tabellen dar. Eine Tabelle besteht aus einer definierten Spaltengruppe und einer beliebigen Anzahl von Zeilen. Die Daten in jeder Tabelle sind logisch miteinander verbunden, und es können Abhängigkeiten zwischen Tabellen definiert werden. Die Daten können auf der Basis mathematischer Grundsätze und Operationen angezeigt und geändert werden, die als Relationen bezeichnet werden (wie INSERT, SELECT und UPDATE).

Eine Datenbank ist selbstbeschreibend in der Hinsicht, dass sie neben den Daten eine Beschreibung ihrer Struktur enthält. Sie umfasst einen Satz von Systemkatalogtabellen, die die logische und physische Struktur der Daten beschreiben, eine Konfigurationsdatei, die die Parameterwerte für die Datenbank enthält, und ein Wiederherstellungsprotokoll, das ablaufende und archivierbare Transaktionen aufzeichnet.

Es gibt lokale und ferne Datenbanken. Eine lokale Datenbank befindet sich physisch auf der verwendeten Workstation, während sich eine ferne Datenbank auf einer anderen Maschine befindet.

Schemata

Ein Schema ist eine eindeutige Kennung zur Gruppierung einer Reihe von Datenbankobjekten (wie Tabellen, Sichten, Indizes und Aliasnamen). Wenn Sie eine Tabelle mit dem Namen PAYROLL erstellen, wäre es mühselig, die Datenbank durchsuchen zu müssen, um festzustellen, ob ein anderer Benutzer bereits eine Tabelle mit diesem Namen erstellt hat. Der Name jedes Objekts muss nur innerhalb seines Schemas eindeutig sein.

Die meisten Datenbankobjekte haben einen zweiteiligen Objektnamen, wobei der erste Teil den Schemanamen und der zweite Teil den Namen des Objekts darstellt. Wenn ein Objekt erstellt wird, können Sie dies einem bestimmten Schema zuordnen. Wenn Sie kein Schema angeben, wird das Objekt dem Standardschema zugeordnet, das in der Regel die Benutzer-ID der Person ist, die das Objekt erstellt hat. Ein Benutzer mit dem Namen Smith kann beispielsweise über eine Tabelle mit dem Namen SMITH.PAYROLL verfügen.

Das Schema wird ebenfalls ein Objekt in der Datenbank. Es wird erstellt, wenn das erste Objekt im Schema erstellt wird. Einzelne Personen können Schemata besitzen, und der Besitzer kann den Zugriff auf die darin enthaltenen Daten und Objekte kontrollieren.

Tabellen

Eine relationale Datenbank ist eine Datensammlung in Form von Tabellen. Eine Tabelle besteht aus Daten, die in Spalten und Zeilen (allgemein als Datensätze bekannt) logisch angeordnet sind.

Jede Tabelle hat einen Namen, und innerhalb einer Tabelle hat jede Spalte einen Namen. Die Zeilen einer Tabelle stehen in keiner bestimmten Reihenfolge, sie können jedoch in einer Reihenfolge abgerufen werden, die durch Werte in ihren Spalten bestimmt wird. Die Daten in einer Tabelle sind logisch miteinander verbunden. Alle Datenbank- und Tabellendaten werden Tabellenbereichen zugeordnet.

Anhang D. Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an

IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France,

zu richten. Anfragen an obige Adresse müssen auf Englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© *(Name Ihrer Firma) (Jahr)*. Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *_Jahr/Jahre angeben_*. Alle Rechte vorbehalten.

Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet:

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
IBM System AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RS/6000
DataPropagator	IBM System /370
DataRefresher	SP
DB2	SQL/400
DB2 Connect	SQL/DS
DB2 Extenders	IBM System/370
DB2 OLAP Server	IBM System /390
DB2 Universal Database	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
iSeries	zSeries

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet.

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken der Microsoft Corporation.

Intel und Pentium sind in gewissen Ländern Marken der Intel Corporation.

Java und alle auf Java basierenden Marken sind in gewissen Ländern Marken von Sun Microsystems, Inc.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Index

A

- Abfrage ohne Indizes und ohne Statistikdaten 14
- Abfrageoptimierungsklassen
 - Definition 47
- Aktuelle Statistikdaten für Tabellen und Indizes erfassen 17, 35
- Aufwand
 - Definition 48
- Ausführen, Abfrage ohne Indizes und ohne Statistikdaten 32

B

- Befehle, EXPLAIN.DDL 1
- Befehle, LIST TABLES 1
- Befehle, Option EXPLSNAP des Befehls BIND 5
- Befehle, SET CURRENT EXPLAIN SNAPSHOT 4
- Befehle, VESAMPL.DDL 2
- Behälter
 - Definition 48
- Benutzerdefinierte Funktionen, Statistikdaten anzeigen 10, 11
- Bindeoptionen, Informationen anzeigen zu 11

C

- CMPEXP, Operator
 - Definition 53, 75
- Cursorblockung
 - Definition 49

D

- Dateien, EXPLAIN.DDL 1
- DELETE, Operator
 - Definition 53, 75
- DMS-Tabellenbereiche
 - Definition 71
- Dynamische SQL-Anweisungen, EXPLAIN-Momentaufnahmen erstellen für 4
- Dynamisches SQL
 - Definition 49

E

- EISCAN, Operator
 - Definition 53, 75
- EXPLAIN.DDL, Datei/Befehl 1

- EXPLAIN-Momentaufnahmen
 - Definition 50
- EXPLAIN-Momentaufnahmen, Beispiele für Visual Explain 2
- EXPLAIN-Momentaufnahmen, erstellen 1
- EXPLAIN-Momentaufnahmen für dynamische SQL-Anweisungen, erstellen 4
- EXPLAIN-Momentaufnahmen für statische SQL-Anweisungen, erstellen 5
- EXPLAIN-Tabellen, erstellen 1
- EXPLSNAP, Option des Befehls BIND 5

F

- FETCH, Operator
 - Definition 54, 76
- FILTER, Operator
 - Definition 54, 76
- Funktionen, Statistikdaten anzeigen 10, 11

G

- GENROW, Funktion
 - Definition 55, 77
- GRPBY, Operator
 - Definition 55, 77

H

- HSJOIN, Operator
 - Definition 56, 78

I

- Indizes
 - Clustering
 - Definition 48
 - Indizes für Spalten erstellen, die zur Verknüpfung von Tabellen in einer Abfrage verwendet werden 22, 40
- INSERT, Operator
 - Definition 57, 79
- Integrierte Funktionen, Statistikdaten anzeigen 10, 11
- IXAND, Operator
 - Definition 57, 79
- IXSCAN, Operator
 - Definition 58, 80

K

- Konfigurationsparameter, Informationen anzeigen zu 11

L

- LIST TABLES, Befehl 1

M

- Mit EXPLAIN bearbeitbare Anweisungen
 - Definition 51
- Mit EXPLAIN bearbeitete Anweisungen
 - Definition 51
- Mit EXPLAIN bearbeitete SQL-Anweisungen, auswählen 7
- Momentaufnahmen, Beispiele für Visual Explain 2
- MSJOIN, Operator
 - Definition 59, 81

N

- NLJOIN, Operator
 - Definition 60, 81

O

- Operanden
 - Definition 51
- Operatoren
 - Definition 51
 - Liste 51
- Operatoren, Details anzeigen zu 10
- Optimierungsprogramm
 - Definition 66

P

- Pakete
 - Definition 66
- PIPE, Operator
 - Definition 61, 82

R

- RETURN, Operator
 - Definition 61, 82
- RIDSCN, Operator
 - Definition 61, 83

S

- Selektivität von Vergleichselementen
 - Definition 67

SET CURRENT EXPLAIN
 SNAPSHOT, Befehl 4
 SHIP, Operator
 Definition 62, 83
 SMS-Tabellenbereiche
 Definition 71
 SORT, Operator
 Definition 62, 83
 Spalten in SQL-Anweisungen,
 Statistikdaten anzeigen 11
 Statische SQL-Anweisungen, EXP-
 LAIN-Momentaufnahmen erstellen
 für 5
 Statisches SQL
 Definition 67
 Statistiken, für Tabellen, Indizes,
 Tabellenfunktionen 9
 Sternverknüpfungen
 Definition 68

Zugriffsplandiagramm, anzeigen
 und verwenden 7
 Zugriffsplandiagramm, Darstellung
 ändern von 11
 Zugriffsplandiagramm, Objekt-
 details 9
 Zugriffsplandiagramm, Symbole ver-
 stehen 8
 Zugriffsplandiagramme
 erstellen
 Definition 73
 Knoten
 Definition 74
 Liste der verwendeten Operato-
 ren 51
 Zugriffspläne
 Definition 72
 Zusätzliche Indizes für Tabellen-
 spalten erstellen 27, 44

T

Tabellenbereiche
 Definition 69
 DMS
 Definition 71
 Tabellenbereiche, Statistikdaten
 anzeigen 11
 TBSCAN, Operator
 Definition 63, 85
 TEMP, Operator
 Definition 64, 86
 TQUEUE, Operator
 Definition 65, 86

U

UNION, Operator
 Definition 65, 86
 UNIQUE, Operator
 Definition 65, 87
 UPDATE, Operator
 Definition 66, 87

V

Vergleichselemente
 Definition 69
 VESAMPL.DDL, Befehl 2
 Visual Explain
 Beschreibung 70

Z

Zeilenblockung
 Cursorblockung 49
 Zoomleiste, zur Vergrößerung eines
 Zugriffsplandiagramms 8
 Zugriffsplan, verbessern 13, 31

Kontaktaufnahme mit IBM

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190/772 243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Informationen zur nächsten IBM Niederlassung in Ihrem Land oder Ihrer Region finden Sie im IBM Verzeichnis für weltweite Kontakte, das Sie im Web unter www.ibm.com/planetwide abrufen können.

Produktinformationen

Informationen zu DB2 Universal Database-Produkten erhalten Sie telefonisch oder im World Wide Web unter www.ibm.com/software/data/db2/udb.

Diese Site enthält die neuesten Informationen zur technischen Bibliothek, zum Bestellen von Büchern, zu Client-Downloads, Newsgroups, FixPaks, Neuerungen und Links auf verfügbare Webressourcen.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können, finden Sie auf der globalen IBM Internet-Seite unter folgender Adresse: www.ibm.com/planetwide

IBM