

DB2 V8

DPSI , MQT , performances

Guy Delaporte
guy_delaporte@fr.ibm.com

Agenda



- Partitionnement
- Index DPSI
- MQT
- Star Join
- Performances
- Recommended in V7
- Query Monitor

DB2. Information Management Software

DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More



SG24-6079

Redbooks

International Technical Support Organization



IBM

Requirement for 4096 Partitions



Installations require more than the current maximum of 254 partitions

- For more granular segments of work, for example, to have partitions for each day for 11 years (that is, needs 4026 partitions)
- More partitions enable smaller partition data set sizes
==> Easier to manage

Maximum number of partitions raised from 254 to 4096

- Table spaces and indexes
- Table space must have LARGE or DSSIZE specified to go beyond 254 parts

Maximum table size remains 16TB for 4 KB pages

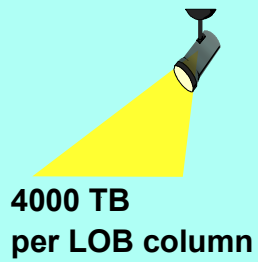
- Can exceed 16 terabytes for a single table with larger page sizes

DB2 Objects Keep Getting Bigger!



128TB
Version 8
Table space

16 TB
Version 6
Table space



1 TB
Version 5
Table space

64 GB Version 4 table space

Creating the Partitioning Index



CREATE INDEX index-name ON CUSTOMER

(ACCOUNT_NUM ASC) ...

CLUSTER

<= required!!

(

PART 1 VALUES (199)

**<= presence of one or
more PART n VALUES
clauses indicates a
partitioning index**

PART 2 VALUES (299)

...

PART 4 VALUES (499)

)

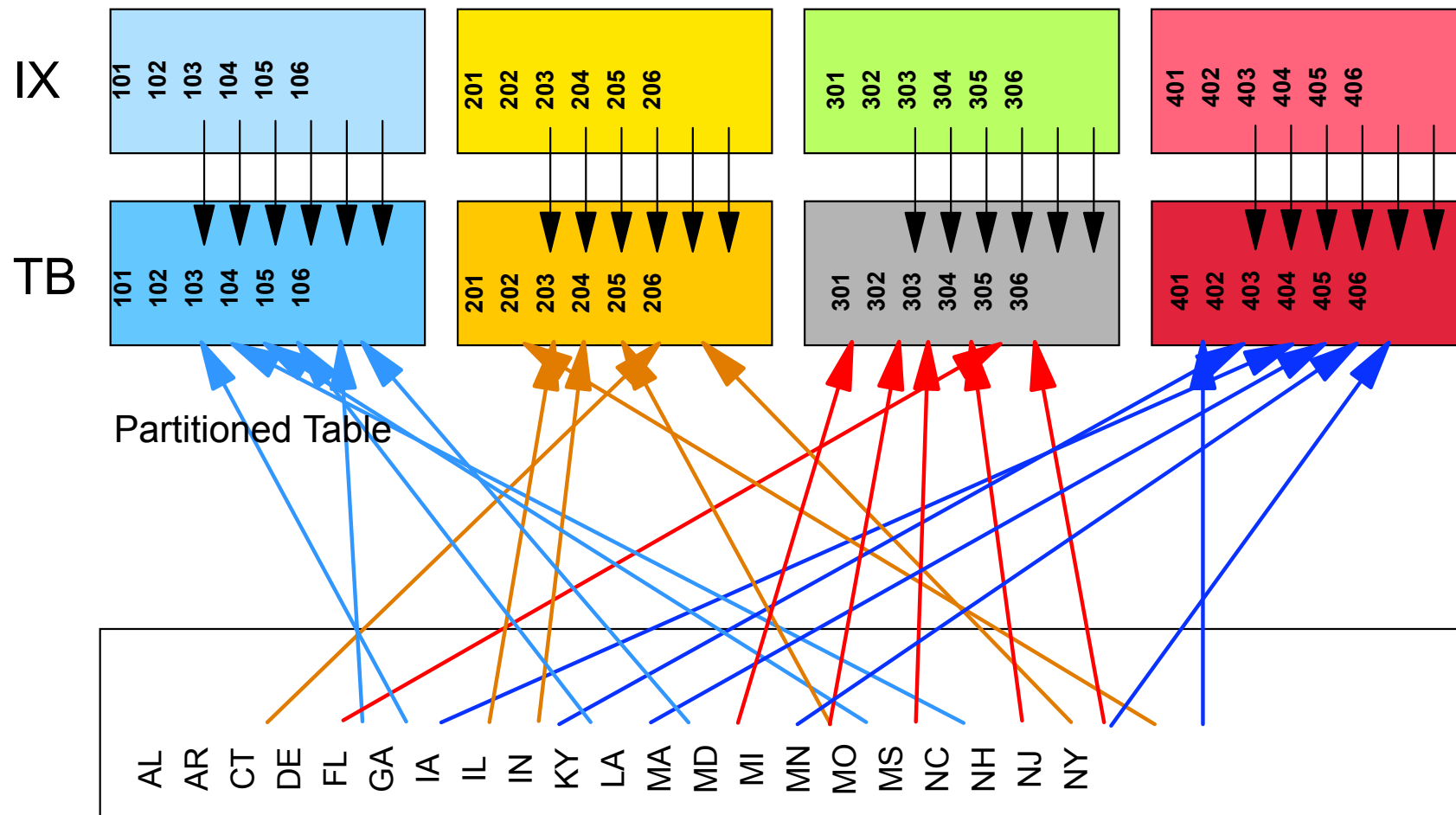
...

**The partitioning index is required to complete the
definition of the partitioned table space!!**

V7- Logical and Physical Partitions



Partitioning and clustering index (on ACCOUNT_NUM) -- both logically and physically partitioned



Partitioned Table

Non-partitioned index (NPI) - Logically partitioned
This index is on STATE_CD

V8 - Creating Partitioned Tables



```
CREATE TABLESPACE tsname NUMPARTS n
(PARTITION 1 USING ...
...
PARTITION n USING ... )           IN dbname;
```

```
CREATE TABLE CUSTOMER (
ACCOUNT_NUM      INTEGER,
CUST_LAST_NM     CHAR(30),
...
LAST_ACTIVITY_DT DATE,
STATE_CD         CHAR(2) )
```

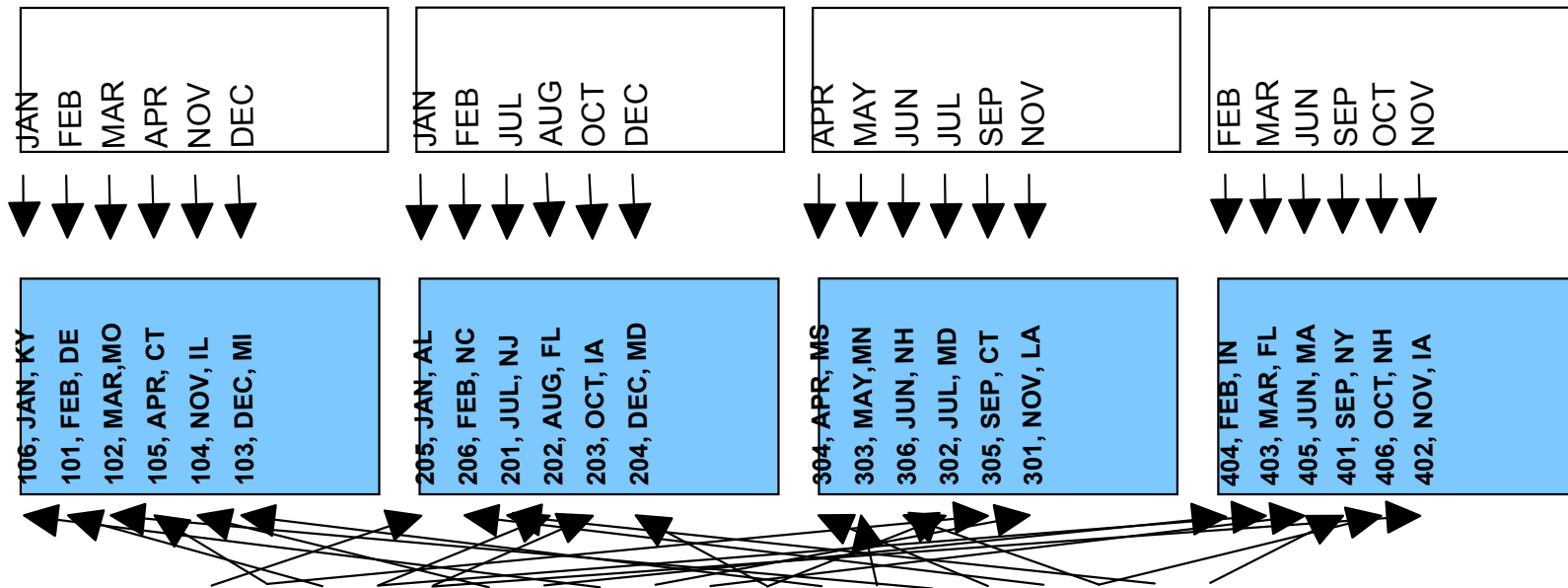
```
PARTITION BY ( ACCOUNT_NUM ASC )
( PARTITION 1 ENDING AT (199),
  PARTITION 2 ENDING AT (299),
  PARTITION 3 ENDING AT (399),
  PARTITION 4 ENDING AT (499) )
IN dbname.tsname;
```

Definition is complete at this point!!

Partitioned and Non-partitioned Secondary Indexes



Data Partitioned Secondary Index (DPSI) -- DPSI1



Partitioned table

Non-Partitioned Secondary Index (NPSI) -- NPSI2

DPSIs and Utility Operations



Data-Partitioned Secondary Index (DPSI)
 DPSI1 on last_activity_date asc

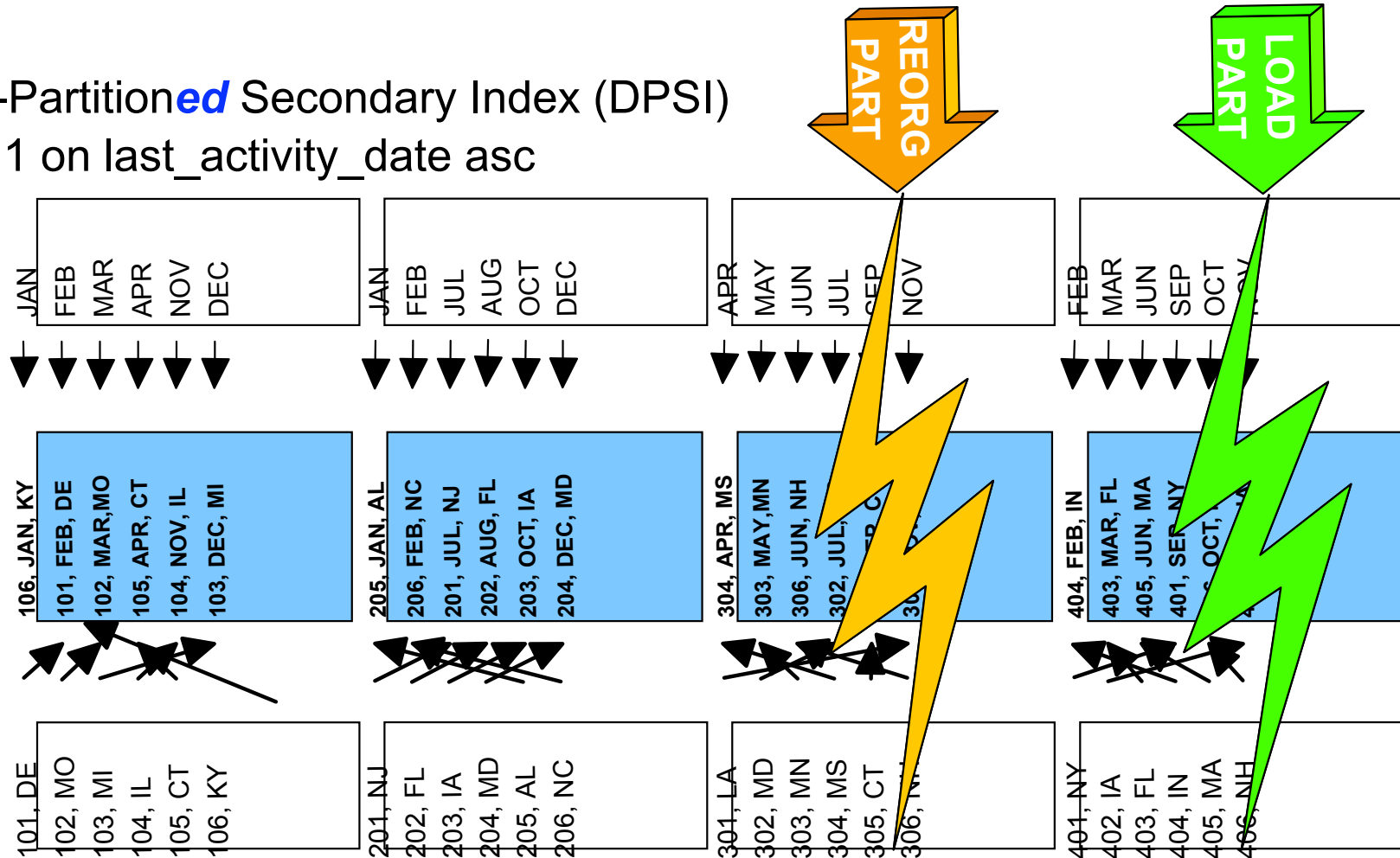


Table Partitioned on account_num, (clustered on last_activity_dt asc)

Data-Partitioned Partitioning Index
 PI2 on account_num asc, state_cd asc

Converting to Table-Controlled Partitioning

No need to DROP/CREATE all existing partitioned tables

DB2 will automatically convert to table-controlled partitioning for you when any of the following SQL statements are executed:

- DROP the partitioning index
- ALTER INDEX NOT CLUSTER on the partitioning index
- ALTER TABLE ... ADD PARTITION
- ALTER TABLE ... ROTATE PARTITION
- ALTER TABLE ... ALTER PARTITION n
- CREATE INDEX ... PARTITIONED
- CREATE INDEX ... ENDING AT ... omitting cluster keyword

Least disruptive approach

- ALTER INDEX xpi NOT CLUSTER of the (current) partitioning index
- ALTER INDEX xpi CLUSTER of the same index

Query performance characteristics of DPSIs

- Allows query parallelism
- Queries with predicates only on secondary index columns will need to scan all partitions
- DB2 tries to do partition pruning -- The application needs to code explicit partitioning key predicates to allow for partition pruning when a DPSI exists
- Chosen for queries with predicates on partitioning columns plus secondary index columns

Secondary indexes

- NPSIs
 - pro: favor sequential query performance
 - con: partition-level query or utility operations
- DPSIs
 - pro: favor partition-level query or utility operation
 - con: sequential query performance, although well-suited to partition parallelism

Schema changes: What is New in V8?



Add a partition to the end of a table

Rotate and rebalance partitions

Unbundling partitioning and clustering

- Ability to drop the partitioning index
- Ability to change the clustering index

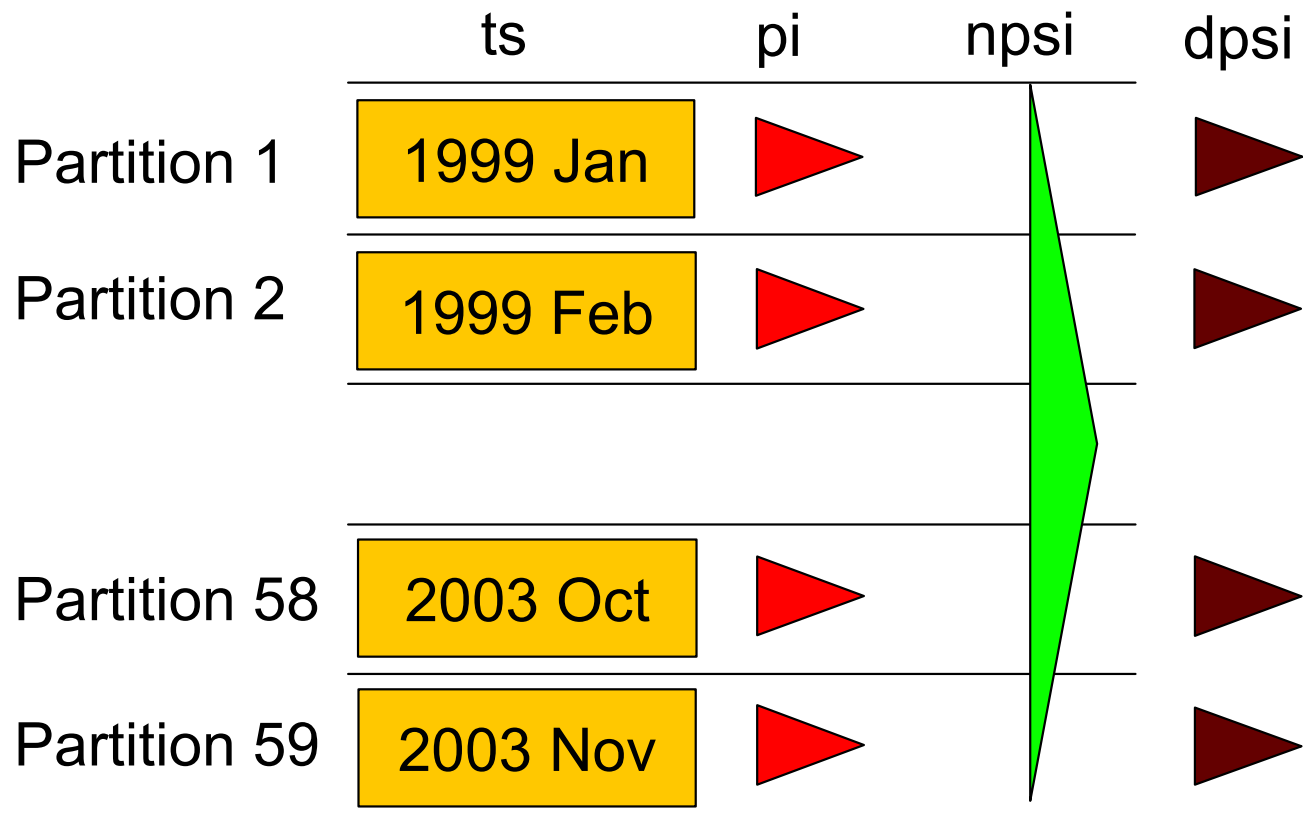
Create or alter an index to have true varying length character columns

Add columns to indexes

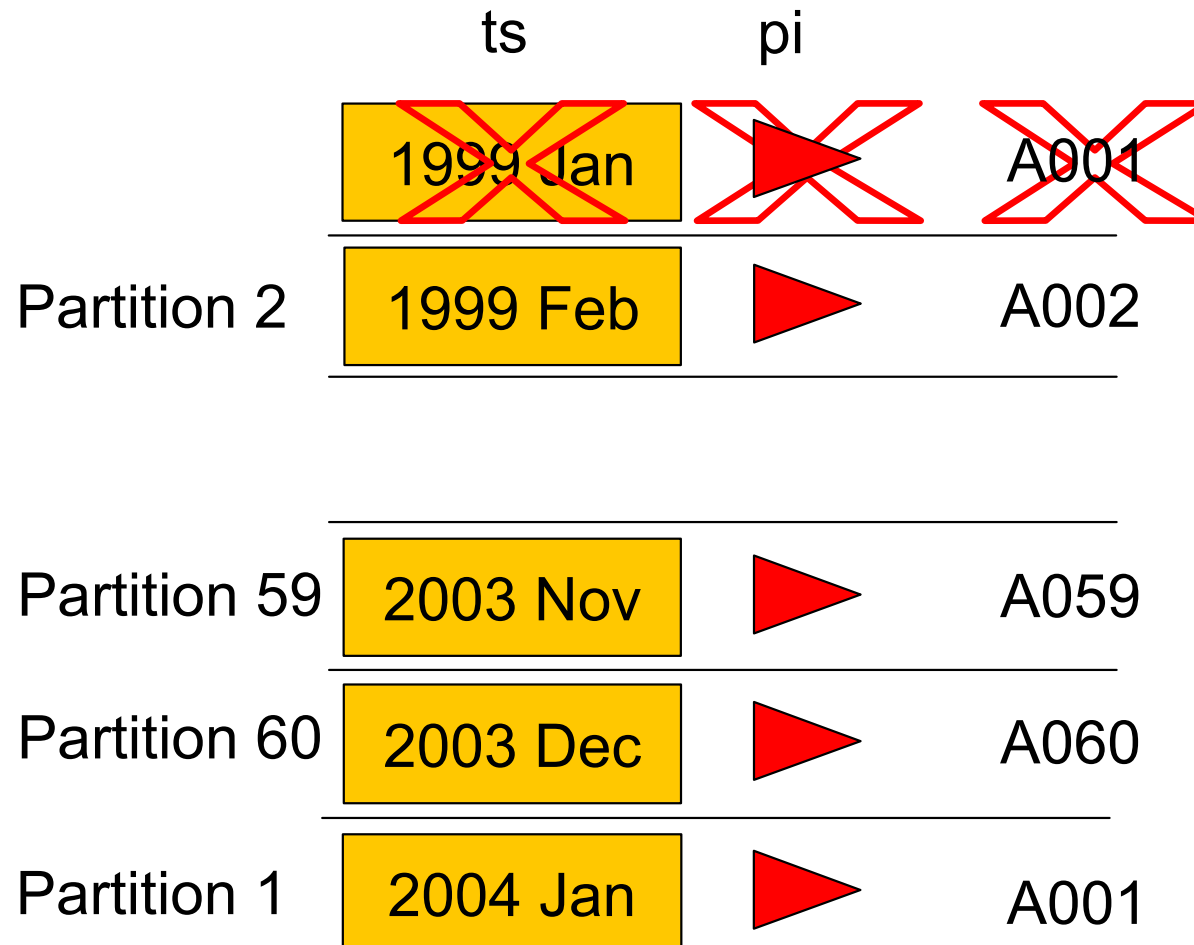
Ability to change data types and lengths (extensions)

- Includes column data types referenced within a view
- Includes column changes for indexed columns

Adding a Table Partition



Rotate Partition Overview



Performance Enhancements

List of Topics



Distribution statistics on non-indexed columns

Materialized query tables

Star Join :Data caching and sparse index

Indexing enhancements

Stage 1 and indexable predicates

Cost-based parallel sort for single and multiple tables

Performance of multi-row operations

Volatile table support

Visual Explain enhancements

RUNSTATS Enhancements



Non-uniform distribution statistics on non-indexed columns

Ability to update statistics history tables with the latest information without updating the statistics used by the optimizer

- Facilitates monitoring and analysis
- RUNSTATS TABLESPACE DB1.TS1

UPDATE NONE HISTORY ALL

RUNSTATS with UPDATE NONE REPORT NO to invalidate dynamic SQL cache without collecting any statistics

The Need for Extra Statistics

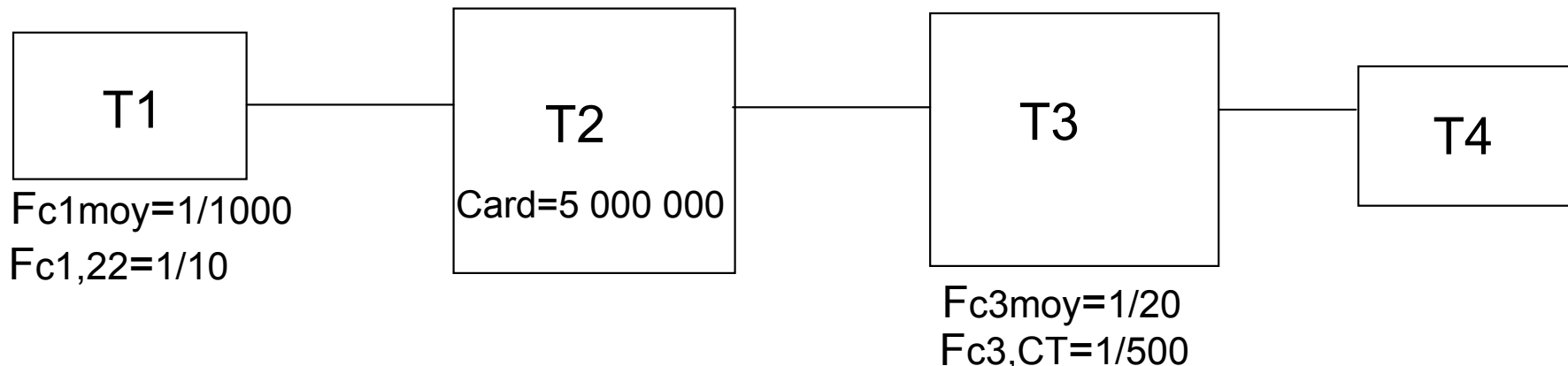


Distribution statistics are currently collected for indexed columns only (if ... `FREQVAL NUMCOLS(1) COUNT(10)`)

Non-uniform distribution statistics for non-leading indexed columns are not collected by RUNSTATS, which can result in non-optimal performance

- Less efficient join sequences
- Inappropriate table join method
- Increase in the number of rows that need to be processed

T1.C1='22' and T3.C3='CT'



RUNSTATS Enhancement



Can collect distribution statistics on any column, or group(s) of columns, indexed and non-indexed, specified at the table level

Frequency distributions for (non-indexed) columns or groups of columns

Cardinality values for groups of (non-indexed) columns

LEAST frequently occurring values, along with MOST for both indexed and non-indexed column distributions

**New keywords : COLGROUP MOST LEAST BOTH
SORTNUM SORTDEVT**

To improve query performance especially for Data Warehousing

Summary tables are often created manually for users

- To improve the response time
- To avoid redundant work of scanning, aggregation and joins of the detailed base tables (e.g. history)
- To simplify SQL to be coded

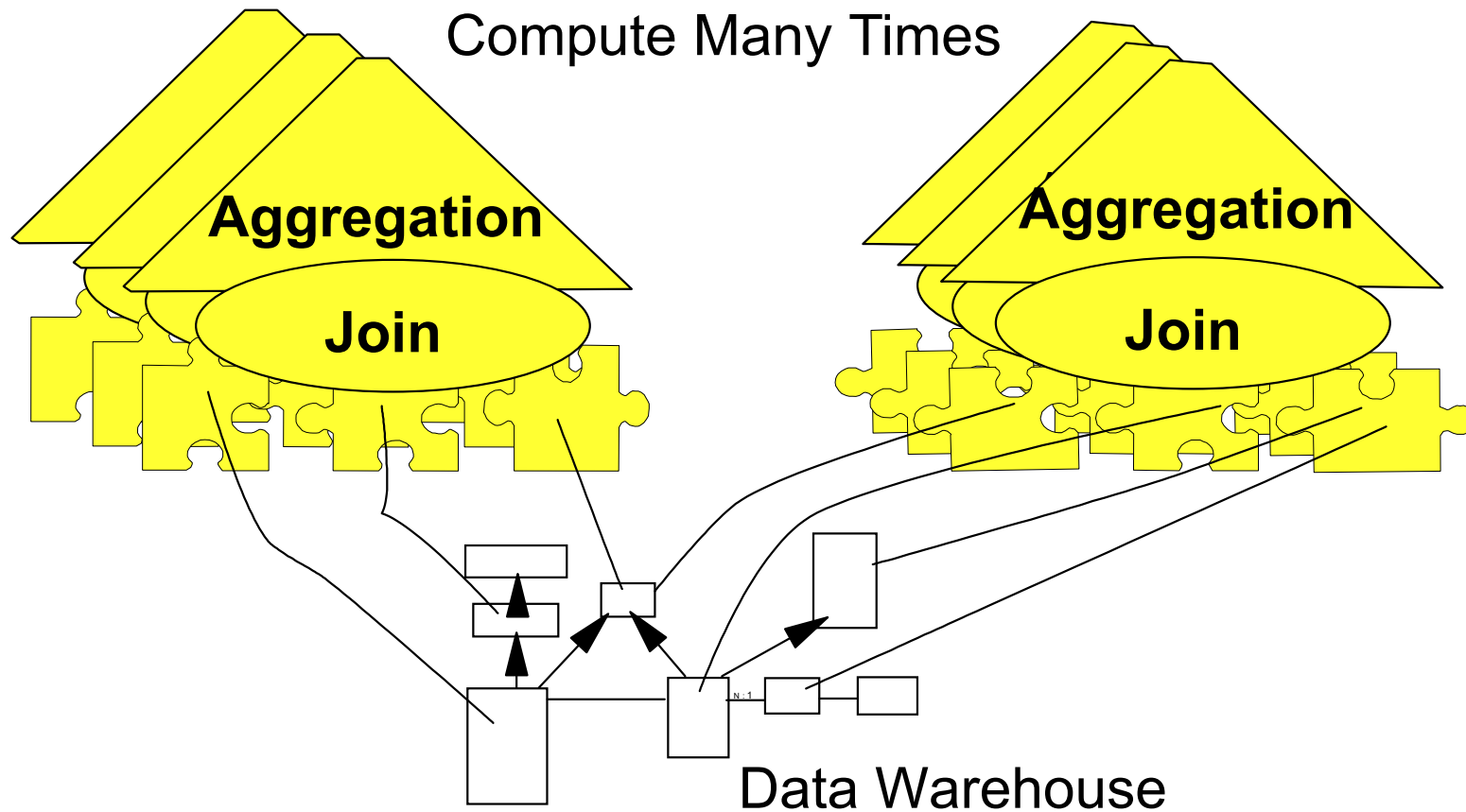
User needs to be aware of summary tables and know whether to use them or the base tables depending on the query.

Without MQT Each Query Re-Computes!

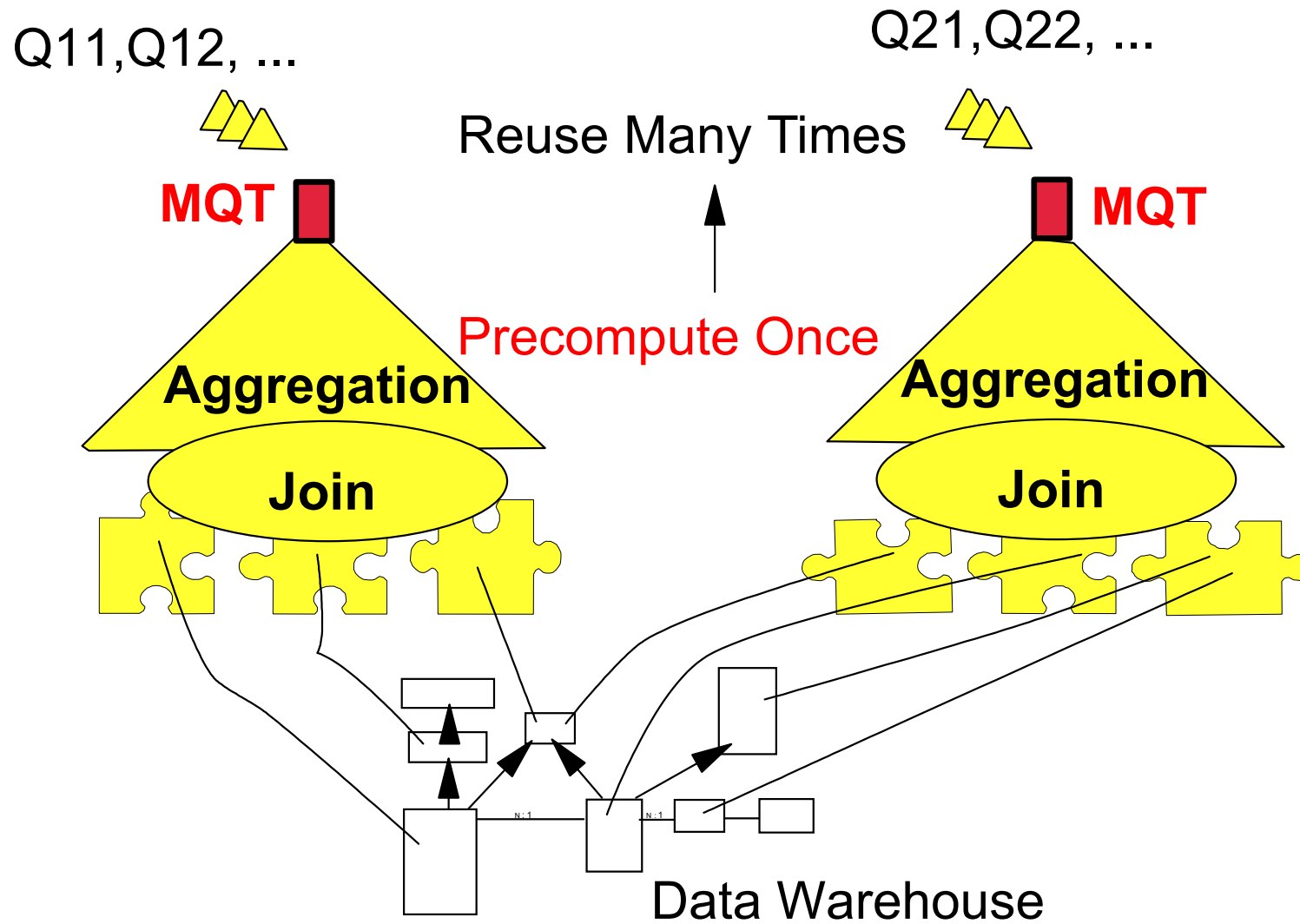


Q11, Q12, ...

Q21, Q22, ...



With MQT Avoid Redundant Computation

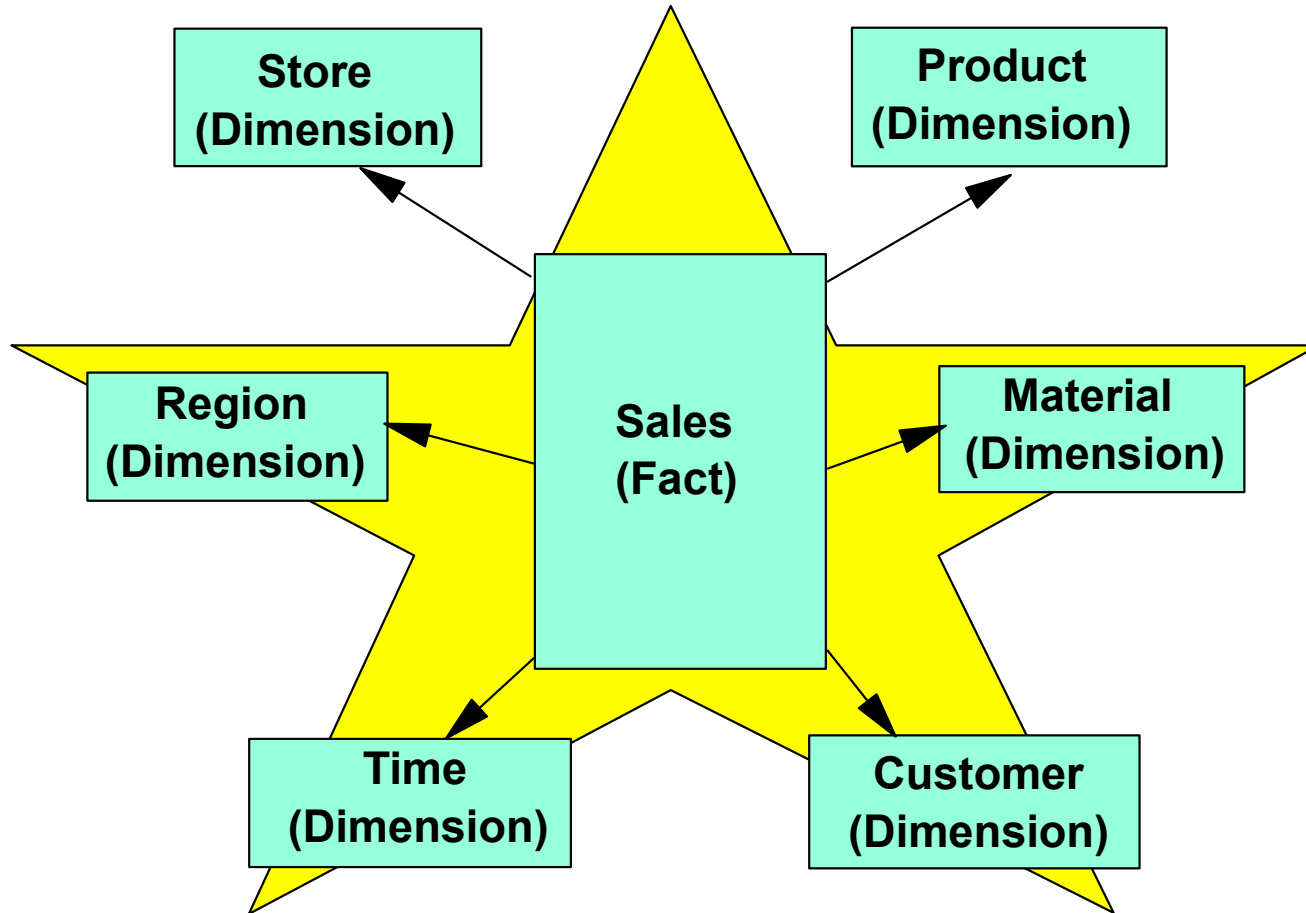


Creating an MQT - Example



```
CREATE TABLE MQT1 AS (  
  SELECT T.PDATE, T.TRANSID,  
         SUM(QTY * PRICE) AS TOTVAL,  
         COUNT(QTY * PRICE) AS CNT  
  FROM SCNDSTAR.TRANSITEM TI, SCNDSTAR.TRANS T  
  WHERE TI.TRANSID = T.TRANSID  
  GROUP BY T.PDATE, T.TRANSID)  
DATA INITIALLY DEFERRED  
REFRESH DEFERRED  
MAINTAINED BY SYSTEM  
ENABLE QUERY OPTIMIZATION  
IN MYDBMQT.MYTSMQT;
```


Star Join Processing Enhancements



Typical Star Schema Query



| id | month | qtr | year |
|----|-------|-----|------|
| 1 | Jan | 1 | 2002 |
| 2 | Feb | 1 | 2002 |
| 3 | Mar | 1 | 2002 |
| 4 | Apr | 2 | 2002 |
| 5 | May | 2 | 2002 |
| 6 | Jun | 2 | 2002 |
| | | | |
| 39 | Mar | 1 | 2003 |

time (dimension)
39 rows

| id | city | region | country |
|----|------------|--------|---------|
| 1 | New York | East | USA |
| 2 | Boston | East | USA |
| 3 | Chicago | East | USA |
| 4 | San Jose | West | USA |
| 5 | Seattle | West | USA |
| 6 | Los Angles | West | USA |

region (dimension)
1,000 rows

| id | item | class | department |
|----|------------|-------|--------------|
| 1 | stereo | audio | audio-visual |
| 2 | cd player | audio | audio-visual |
| 3 | television | video | audio-visual |
| | | | |

product (dimension)
60,000 rows

| time | locn | prod | customer | seller | |
|------|------|------|----------|--------|------|
| 1 | 1 | 1 | 123 | 22 | |
| 2 | 5 | 2 | 345 | 33 | |
| 2 | 2 | 2 | 567 | 66 | |
| 2 | 2 | 1 | 789 | 12 | |
| 3 | 3 | 3 | 112 | 23 | |
| 3 | 6 | 2 | 348 | 78 | |
| 2 | 6 | 1 | 777 | 60 | |

sales (fact)
150 million rows

```

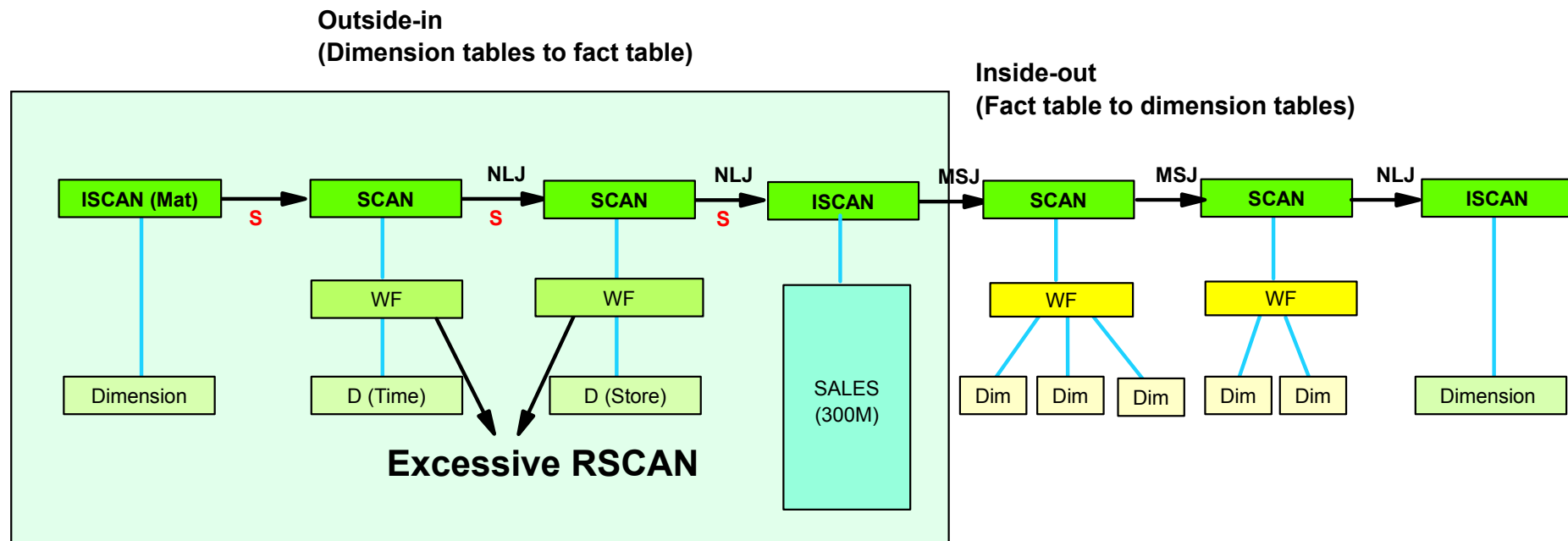
SELECT *
FROM SALES S, TIME T,
      REGION L, PRODUCT P
WHERE S.TIME = T.ID
      AND S.REGION = L.ID
      AND S.PRODUCT = P.ID
      AND T.YEAR = 2002
      AND T.QTR = 1
      AND L.CITY IN ('Boston','Seattle')
      AND P.ITEM = 'stereo';
    
```

Workfiles in Outside-In Processing



During 'logical' cartesian product, DB2 frequently repositions in the dimension workfiles

Processing of the workfiles can only be done by scanning them over and over again (RSCAN)



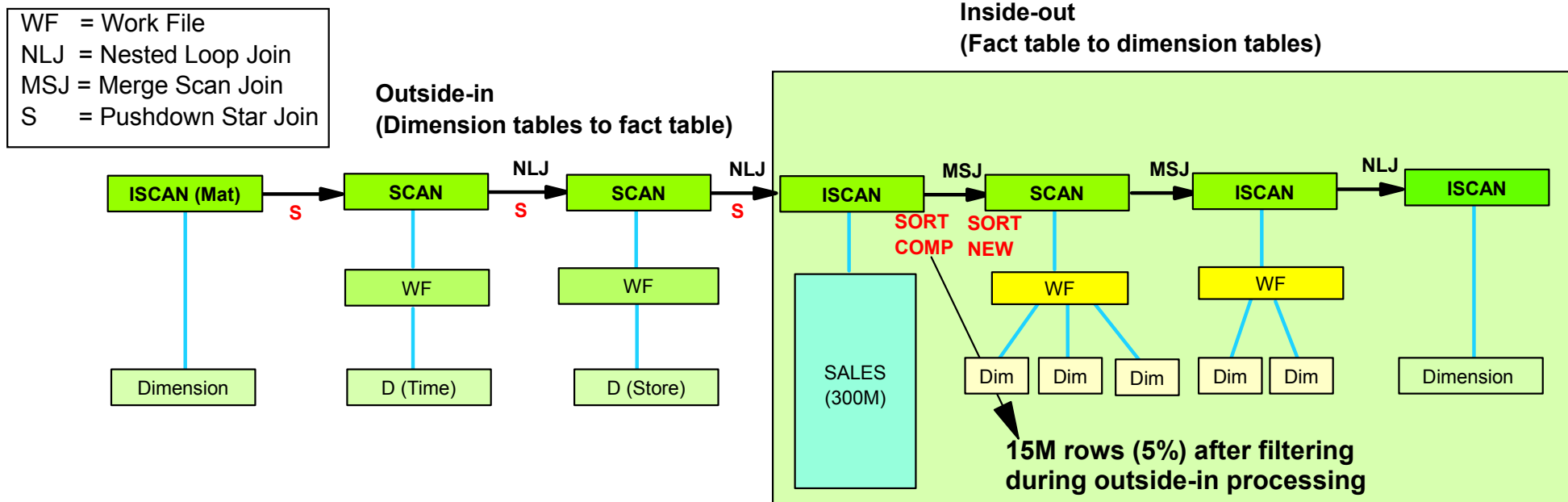
Inside-Out Stage Workfile Challenges

Typical data warehouse queries touch significant portion of fact table

In the inside-out stage, sort is performed on the composite table (the result of outside-in processing) at the start of MSJ processing (materialized work file has no index)

Sorting the large result set is a performance concern (time and space)

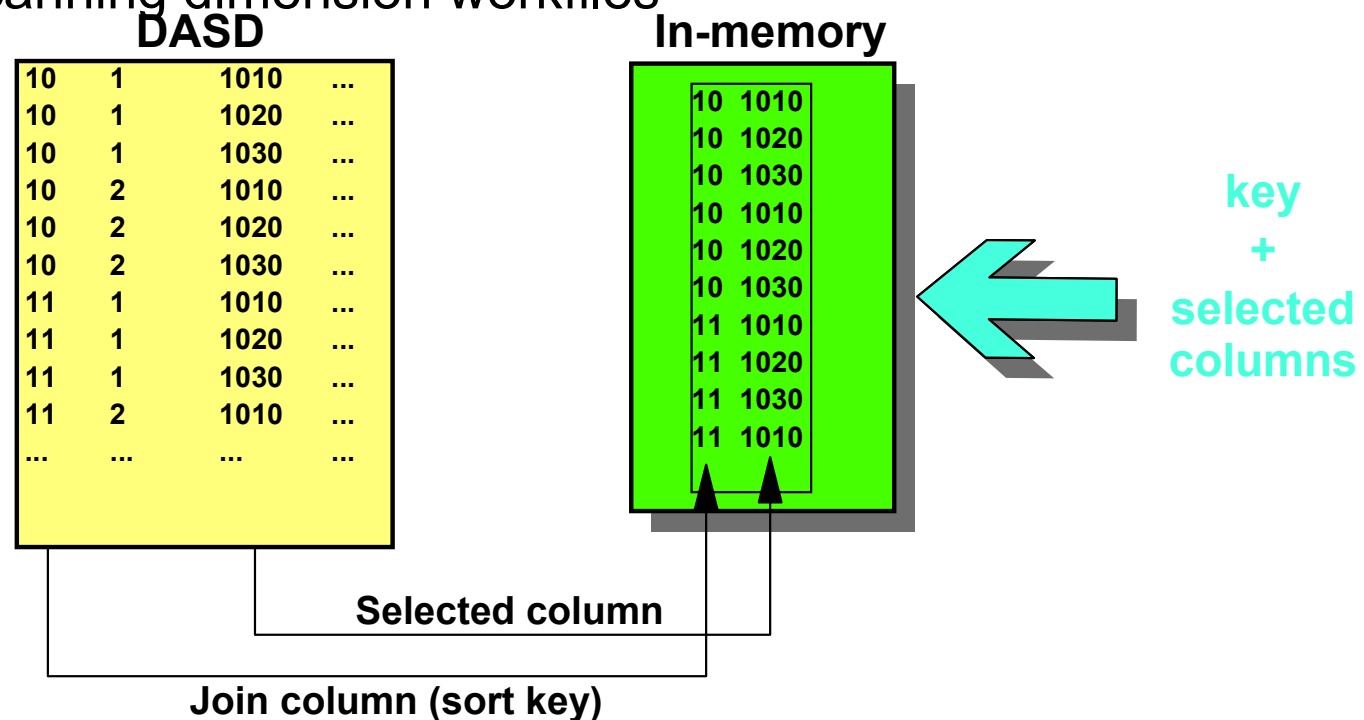
Sort of "new" table (materialized snowflake - workfile) before MSJ



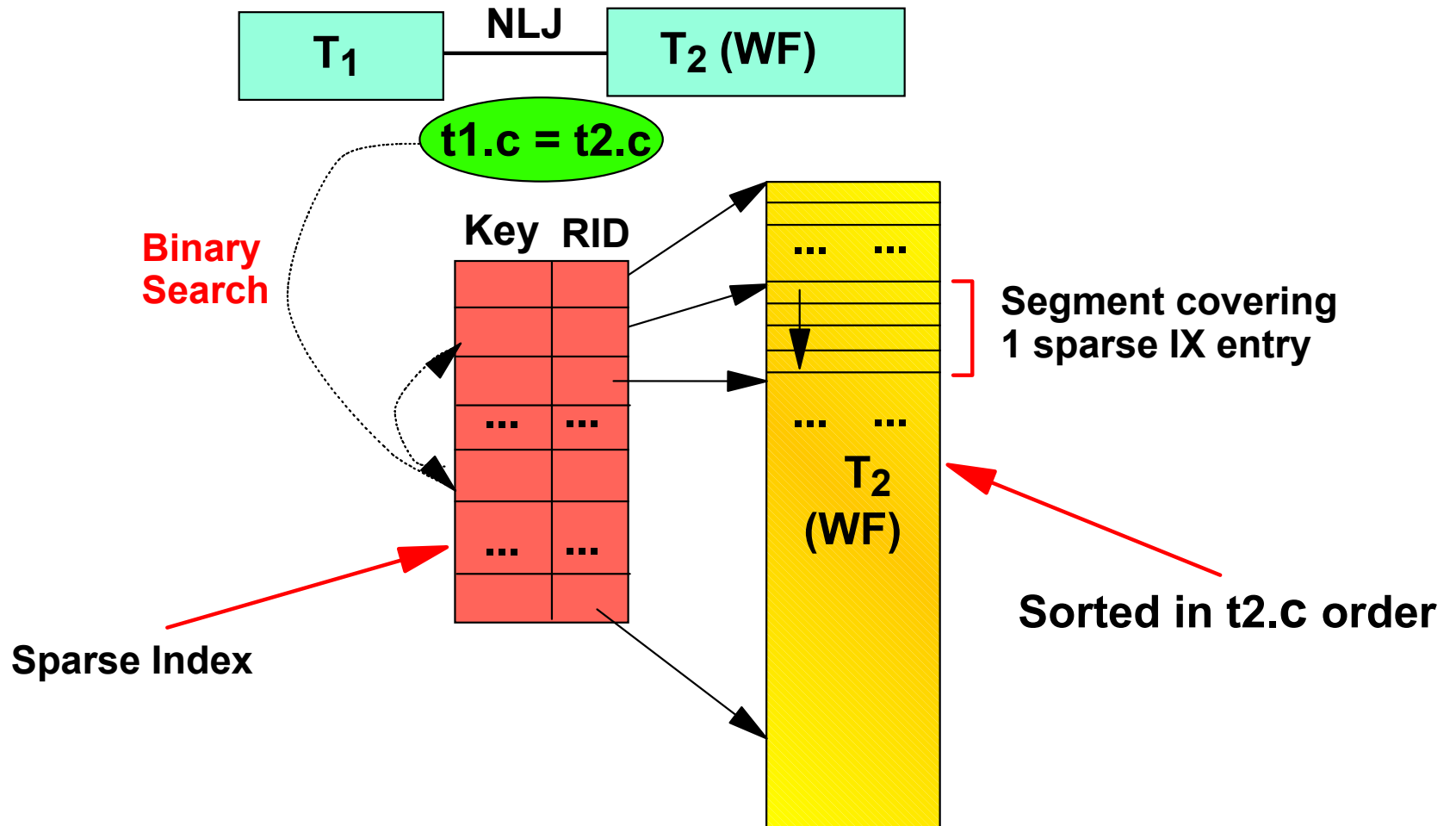
In-memory Workfiles



- In-memory data structure - above the bar
- Sorted in the join column order
- Containing only the join column and the selected columns
- Binary search for the target row
- More beneficial for large join composite
- Ideal for scanning dimension workfiles



How Does Sparse Index Work?



Considérations V7 et Performances

✓ **Statistiques non uniformes**

- DSTATS pour toute colonne d'une table
Download depuis
<http://www-1.ibm.com/support/docview.wss?uid=swg24001598>
- KEYCARD pour corrélation de colonnes d'index
- FREQVAL NUMCOLS(1) COUNT(10)
pour distribution non uniforme de colonne présente dans index

✓ **Star join**

- DSNZPARAM (STARJOIN(DISABLE) , SJTABLES(10))

✓ **et les PTF récentes (PQ65335 , PQ80647, OPTCCOS1, ...)**

✓ QUERY MONITOR

- Consommations précises par ordre SQL
- Hit-parade actualisé des packages / ordres SQL consommateurs
- Faible overhead pour collecte permanente

✓ Visual Explain V8

- Compatible DB2 V7
- Download depuis
<http://www-306.ibm.com/software/data/db2/zos/osc/ve/beta/index.html>



2004/11/17 16:26:04 ----- **Operational Summaries** ----- Row 1 of 9
 DB2 QM Subsystem: QMV2 Interval Start Date: 11/17/2004 Time: 15:28:02
 Interval End Date: CURRENT Time: CURRENT

DB2: Plan: Pgm: Authid: CorrID:
 Section: Call: Type:
 WSUser: WSName:
 WSTran:

C:2-DB2,P-Plan,R-Pgm,U-Authid,I-Corr,T-Sect,C-Call,W-WSUser,M-WSName,N-WSTran
 S-SQL,O-Objs,D-Delay,B-Buff,L-Lock

| CMD | PLAN | Calls | Elapsed | %Elap | Avg Elap | CPU |
|-----|----------|--------|--------------|-------|----------|-------------|
| --- | --- | --- | --- | --- | --- | --- |
| --- | DSNREXX | 145720 | 10:56.943954 | 87.96 | 0.00 | 1:52.079281 |
| --- | DB2PM | 142428 | 32.002086 | 4.28 | 0.00 | 18.241964 |
| --- | COMPLAN1 | 3252 | 3.404778 | 0.45 | 0.00 | 1.017917 |
| --- | FPEPLAN | 11618 | 11.485900 | 1.53 | 0.00 | 1.584305 |
| --- | ADB | 1263 | 19.858834 | 2.65 | 0.01 | 0.754205 |
| --- | ETIPLAN1 | 288 | 4.083918 | 0.54 | 0.01 | 0.296414 |
| --- | HAAPLAN2 | 632 | 1.255293 | 0.16 | 0.00 | 0.148109 |
| --- | HAAPLAN1 | 1106 | 1.351775 | 0.18 | 0.00 | 0.137069 |
| --- | DISTSERV | 198 | 16.443511 | 2.20 | 0.08 | 0.067735 |

***** Bottom of Data *****

Option ==>

F1=HELP
F7=UP

F2=SPLIT
F8=DOWN

F3=END
F9=SWAP

F4=PREV
F10=LEFT

F5=IFIND
F11=RIGHT

Scroll ==> CSR

F6=NEXT
F12=RETRIEVE





2004/11/17 16:28:12 ----- **Operational Summaries** ----- Row 1 of 20
 DB2 QM Subsystem: QMV2 Interval Start Date: 11/17/2004 Time: 15:28:02
 Interval End Date: CURRENT Time: CURRENT

DB2: Plan: DSNREXX Pgm: Authid: CorrID:
 Section: Call: Type:
 WSUser: WSName:
 WSTran:

C:2-DB2,P-Plan,R-Pgm,U-Authid,I-Corr,T-Sect,C-Call,W-WSUser,M-WSName,N-WSTran
 S-SQL,O-Objs,D-Delay,B-Buffer,L-Lock

| CMD | Program | Calls | Elapsed | %Elap | Avg Elap | CPU | %CPU |
|-----|---------|-------|-------------|-------|----------|-----------|-------|
| --- | --- | --- | --- | --- | --- | --- | --- |
| --- | EMPADD2 | 22198 | 2:22.576450 | 21.70 | 0.00 | 20.575695 | 18.35 |
| --- | SFNSEL2 | 11099 | 34.831792 | 5.30 | 0.00 | 17.803489 | 15.88 |
| --- | JBSSEL2 | 10814 | 31.599573 | 4.81 | 0.00 | 15.215631 | 13.57 |
| --- | DPTSEL2 | 9394 | 28.755072 | 4.37 | 0.00 | 14.875274 | 13.27 |
| --- | PRJADD2 | 38652 | 45.111955 | 6.86 | 0.00 | 10.671171 | 9.52 |
| --- | DSNREXX | 8624 | 5:05.386241 | 46.48 | 0.03 | 9.469639 | 8.44 |
| --- | EMPSEL2 | 32780 | 11.849310 | 1.80 | 0.00 | 6.268592 | 5.59 |
| --- | DPTDEL2 | 109 | 9.381566 | 1.42 | 0.08 | 4.339901 | 3.87 |
| --- | EMPAN02 | 4036 | 9.182500 | 1.39 | 0.00 | 3.574849 | 3.18 |
| --- | LCNSEL2 | 1548 | 4.647511 | 0.70 | 0.00 | 2.454965 | 2.19 |
| --- | DPTADD2 | 1033 | 11.245040 | 1.71 | 0.01 | 1.327247 | 1.18 |
| --- | DPTUPR2 | 606 | 5.156943 | 0.78 | 0.00 | 0.951482 | 0.84 |
| --- | DPTMGR2 | 1296 | 3.524794 | 0.53 | 0.00 | 0.892293 | 0.79 |
| --- | PRJSEL2 | 444 | 1.178248 | 0.17 | 0.00 | 0.782257 | 0.69 |
| --- | DPTAN02 | 828 | 1.997709 | 0.30 | 0.00 | 0.728400 | 0.64 |
| --- | EMPUPD2 | 1024 | 2.909617 | 0.44 | 0.00 | 0.628310 | 0.56 |
| --- | DPTUPD2 | 375 | 3.535546 | 0.53 | 0.00 | 0.598002 | 0.53 |

Option ==> Scroll ==> CSR
 F1=HELP F2=SPLIT F3=END F4=PREV F5=IFIND F6=NEXT
 F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE





2004/11/17 16:11:35 ---- Activity by SQL Text ---- Row 1 of 13
 DB2 QM Subsystem: QMV2 Interval Start Date: 11/17/2004 Time: 15:28:02
 Interval End Date: CURRENT Time: CURRENT
 DB2: Plan: DSNREXX Pgm: EMPADD2 Authid: CorRID:
 Section: Call: Type:
 WSUser: WSName:
 WSTran:

C:0-Objs,D-Delay,B-Buffer Pool,L-Lock,A-Analyze,V-View

| CMD | SQL Text | Calls | DB2 | Plan | Program |
|----------------------------|--|-------|------|---------|---------|
| --- | --- | --- | --- | --- | --- |
| --- | INSERT INTO WLDTEMP (EMP_NO , FIRSTNME | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT INT (: H * RAND () + 0.5) | 1796 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT : H CONCAT SUBSTR (: H , | 1796 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT CURRENT DATE - (INT (: H * | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT : H + 1 INTO : H FROM SYS | 1796 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT CURRENT DATE - (7305 + : H) | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT INT (16436 * RAND ()) INTO : H | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT LENGTH (: H) INTO : H F | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | SELECT ' ' INTO : H FROM SYSIBM . SYS | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | CALL SFNSEL2 (: H , : H , : H | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | CALL DPTSEL2 (: H , : H , : H | 449 | DBDM | DSNREXX | EMPADD2 |
| --- | CALL EMPAN02 (: H , : H) | 450 | DBDM | DSNREXX | EMPADD2 |
| --- | CALL JBSSEL2 (: H , : H , : H | 449 | DBDM | DSNREXX | EMPADD2 |
| ***** Bottom of Data ***** | | | | | |

Option ==> Scroll ==> CSR
 F1=HELP F2=SPLIT F3=END F4=PREV F5=IFIND F6=NEXT
 F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE





```

2004/11/17 16:32:30 ---- Activity by SQL Text ---- Row 1 of 13
DB2 QM Subsystem: QMV2 Interval Start Date: 11/17/2004 Time: 15:28:02
Interval End Date: CURRENT Time: CURRENT
DB2: Plan: DSNREXX Pgm: EMPADD2 Authid: CorRID:
Section: Call: Type:
WSUser: WSName:
WSTran:

```

C:0-Objs,D-Delay,B-Buffer Pool,L-Lock,A-Analyze,V-View

| CMD | Sect | £ | Collection | Elapsed | %Elap | Avg Elap | CPU |
|-----|------|---|------------|-----------|-------|----------|----------|
| — | 18 | | DMDELAP | 8.258319 | 5.79 | 0.00 | 4.121182 |
| — | 11 | | DMDELAP | 3.943339 | 2.76 | 0.00 | 1.913281 |
| — | 12 | | DMDELAP | 3.655375 | 2.56 | 0.00 | 1.905695 |
| — | 16 | | DMDELAP | 3.591336 | 2.51 | 0.00 | 1.761752 |
| — | 13 | | DMDELAP | 3.957290 | 2.77 | 0.00 | 1.719903 |
| — | 15 | | DMDELAP | 3.099894 | 2.17 | 0.00 | 1.717190 |
| — | 14 | | DMDELAP | 3.191876 | 2.23 | 0.00 | 1.618862 |
| — | 9 | | DMDELAP | 3.435504 | 2.40 | 0.00 | 1.556128 |
| — | 10 | | DMDELAP | 3.074151 | 2.15 | 0.00 | 1.447839 |
| — | 3 | | DMDELAP | 39.293343 | 27.55 | 0.03 | 1.331931 |
| — | 4 | | DMDELAP | 22.568871 | 15.82 | 0.02 | 1.133577 |
| — | 2 | | DMDELAP | 10.468595 | 7.34 | 0.01 | 0.194061 |
| — | 17 | | DMDELAP | 34.038553 | 23.87 | 0.03 | 0.154286 |

***** Bottom of Data *****

```

Option ==>
F1=HELP      F2=SPLIT    F3=END      F4=PREV    F5=IFIND    F6=NEXT
F7=UP        F8=DOWN     F9=SWAP     F10=LEFT   F11=RIGHT   F12=RETRIEVE
Scroll ==> CSR

```





2004/11/17 16:14:00 ---- Display SQL Statement Text ---- Row 1 of 5

DB2: Plan: DSNREXX Pgm: EMPADD2 AuthID: CorrID:
Section: Call: Type:

```
INSERT INTO WLDTEMP ( EMP_NO , FIRSTNME , MIDINIT , LASTNAME , WORKDEPT
, PHONENO , HIREDATE , JOB , MANAGER , EDLEVEL , SEX , BIRTHDATE ,
SALARY , BONUS , COMM , CREATED_BY , UPDATED_BY ) VALUES ( : H , : H , :
H , : H , : H , : H , : H , 'N' , : H , : H , : H , : H , : H , :
H , USER , USER )
```

***** Bottom of Data *****

Option ==> _____ Scroll ==> CSR
F1=HELP F2=SPLIT F3=END F4=PREV F5=IFIND F6=NEXT
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

MA

a

30/015

Connected to remote server/host UKCEMVS1.GFD.UK.IBM.COM using port 3004





2004/11/17 16:33:35 ----- Buffer Pool Statistics ----- Row 1 of 16

DB2 SSID: Plan: DSNREXX DBRM: EMPADD2 Coll: DMDELAP
Cursor: Section:

```
-----
```

| Buffer Pool: ALL | Total | Average |
|--------------------------------|--------|---------|
| Buffer Pool Hit Ratio (%) | N/A | 99.80 |
| Hiper Pool Hit Ratio (%) | N/A | 100.00 |
| Get Page Requests | 49,766 | 49.32 |
| Buffer Pages Updated | 10,365 | 10.27 |
| Synchronous Pages Read | 43 | 0.04 |
| Synchronous Pages Written | 5 | 0.00 |
| Sequential Prefetch Requests | 1,009 | 1.00 |
| List Prefetch Requests | 0 | 0.00 |
| Dynamic Prefetch Requests | 2 | 0.00 |
| Successful Hiper Pool Reads | 0 | 0.00 |
| Hiper Pool Read Failures | 0 | 0.00 |
| Successful Hiper Pool Writes | 0 | 0.00 |
| Unsuccessful Hiper Pool Writes | 0 | 0.00 |
| Async Pages Read | 57 | 0.05 |
| Async Pages Read by Hiper Pool | 0 | 0.00 |

***** Bottom of Data *****

Option ==> _____ Scroll ==> CSR

| | | | | | |
|---------|----------|---------|----------|-----------|--------------|
| F1=HELP | F2=SPLIT | F3=END | F4=PREV | F5=IFIND | F6=NEXT |
| F7=UP | F8=DOWN | F9=SWAP | F10=LEFT | F11=RIGHT | F12=RETRIEVE |





```

2004/11/17 16:15:48 ----- Object Detail ----- Row 1 of 38
DB2 QM Subsystem: QMV2      Interval Start Date: 11/17/2004  Time: 15:28:02
                             Interval End   Date: CURRENT      Time: CURRENT
DB2: DBDM      Plan: DSNREXX  Pgm: EMPADD2  AuthID:      CorrID:
      Section:      18 Call:      Type:
      WSUser:      WSName:
      WSTran:
    
```

```

-----+>
Creator      Name                Type      DataBase  BPool     PageSet    GetPages    HitRatio
-----
SYSIBM      DSNSPT01             INDEX     DSNDB01   BP0       DSNSPT01   27          100.00
SYSIBM      SPTR                 TABLE    DSNDB01   BP0       SPT01      16          0.00
SYSIBM      DSN DYX01            INDEX     DSNDB06   BP0       DSN DYX01  1700       100.00
SYSIBM      DSN DTX01            INDEX     DSNDB06   BP0       DSN DTX01  1706       100.00
SYSIBM      SYSTABLES           TABLE    DSNDB06   BP0       SYSDBASE   1703       100.00
SYSIBM      DSN DSX01            INDEX     DSNDB06   BP0       DSN DSX01  1704       100.00
SYSIBM      SYSTABLESPACE       TABLE    DSNDB06   BP0       SYSDBASE   852        100.00
SYSIBM      SYSINDEXES          TABLE    DSNDB06   BP0       SYSDBASE   5102       100.00
SYSIBM      DSN TTX01            INDEX     DSNDB06   BP0       DSN TTX01  4250       100.00
SYSIBM      DSN DXX01            INDEX     DSNDB06   BP0       DSN DXX01  1700       100.00
SYSIBM      DSN ATX02            INDEX     DSNDB06   BP0       DSN ATX02  2559       100.00
SYSIBM      DSN OTX02            INDEX     DSNDB06   BP0       DSN OTX02  1702       100.00
SYSIBM      SYSTRIGGERS         TABLE    DSNDB06   BP0       SYSOBJ     850        100.00
DMDELAP     WLDXEMP1             INDEX     DMDELAP   BP16      WLDXEMP1   1792       99.11
DMDELAP     WLDTEMP             TABLE    DMDELAP   BP15      WLDSEMP    2550       100.00
DMDELAP     WLDXEMP2            INDEX     DMDELAP   BP16      WLDXEMP2   1708       100.00
DMDELAP     WLDXEMP3            INDEX     DMDELAP   BP16      WLDXEMP3   1710       99.77
DMDELAP     WLDXEMP4            INDEX     DMDELAP   BP16      WLDXEMP4   1708       100.00
N/A         N/A                 TABLE    DSNDB07   BP7       DSN4K03    856        100.00
    
```

```

Option ==>
F1=HELP      F2=SPLIT      F3=END        F4=PREV      F5=IFIND      F6=NEXT
F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT     F12=RETRIEVE
Scroll ==> CSR
    
```



Merci
Questions ?