Information On Demand

SQL Performance Tuning in DB2 9

Mike Bracey, Systems Engineer IBM UK
Based on material developed by:
Patrick Bossman, Senior SW Engineer, IBM US
Brian Baggett, Senior SW Engineer, IBM US

ON DEMAND BUSINESS

Information On Demand

Agenda

- Introduction to
  Optimization Service Center (OSC)
  and
  Optimization Expert (OE)
- Tuning workloads
- Individual query analysis and tuning
- Summary

Overview

- Optimization Service Center
  - Workstation tool for monitoring and tuning of queries
  - Facilitates the identification and tuning of workloads (sets of queries) as well as individual queries
  - New powerful query diagnostic tools enable faster deep analysis of queries
- Optimization Expert
  - All of Optimization Service Center functionality
  - Plus… extra advisors
    - Index advisor, access path advisor, query advisor

OSC and OE

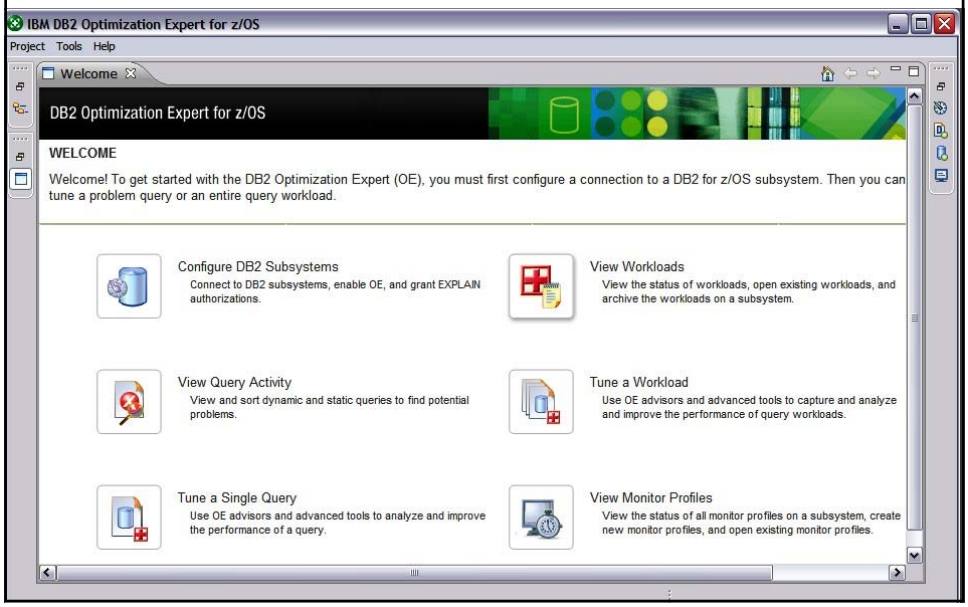
- DB2 for z/OS Optimization Service Center (OSC)
  - New no-charge offering
  - Included in DB2 Accessories Suite for z/OS V1.1
  - Available via web download
- DB2 Optimization Expert for z/OS V1.2 (OE)
  - New One Time Charge (OTC) offering
  - Value Unit, Reference based pricing
- Simultaneous availability with DB2 9 for z/OS
  - GA in March 2007

VE, OSC, OE feature comparison

Functions	Visual Explain	Optimization Service Center	Optimization Expert
Queries from Cache, Catalog	Yes	Yes	Yes
Query Formatter, Annotation		Yes	Yes
Access Plan Graph	Yes	Yes	Yes
Visual Plan Hint		Yes	Yes
Query Statistics Advisor	Yes	Yes	Yes
Workload Statistics Advisor		Yes	Yes
Profile based Monitoring		Yes	Yes
Query Index Advisor			Yes
Workload Index Advisor			Yes
Query Advisor			Yes
Workload Query Advisor			Yes
Access Path Advisor			Yes

4

Welcome page



Tuning an application

Application tuning process

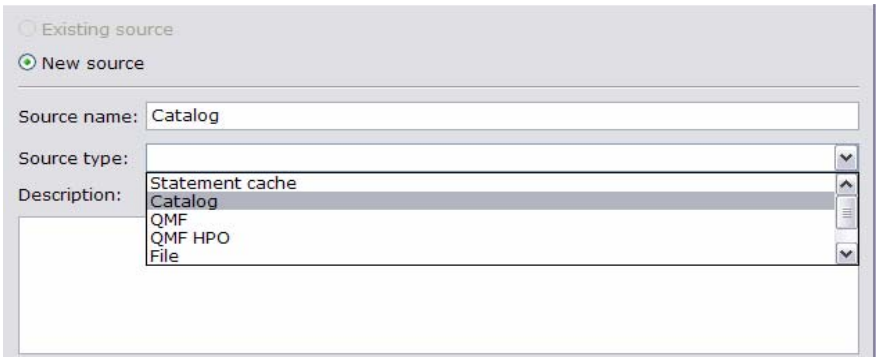
- What is application tuning?
  - Identify what the application workload is
    - Individual SQL statements
  - Identify workload characteristics
    - How often are individual SQL statements executing?
    - What is the performance of individual SQL statements?
  - Build sound optimization foundation using workload advisors
    - Workload statistics advisor to ensure adequate statistics
    - Workload index advisor to ensure adequate indexing
    - Workload query advisor to review SQL
  - Remeasure workload
    - Identify top tuning candidates
    - Use query based tools to further analyze

Generating a workload

- Workload sources
  - Snap statement cache
  - Catalog (static SQL)
  - QMF / QMF HPO
  - File
  - Other workloads

8

Workload sources



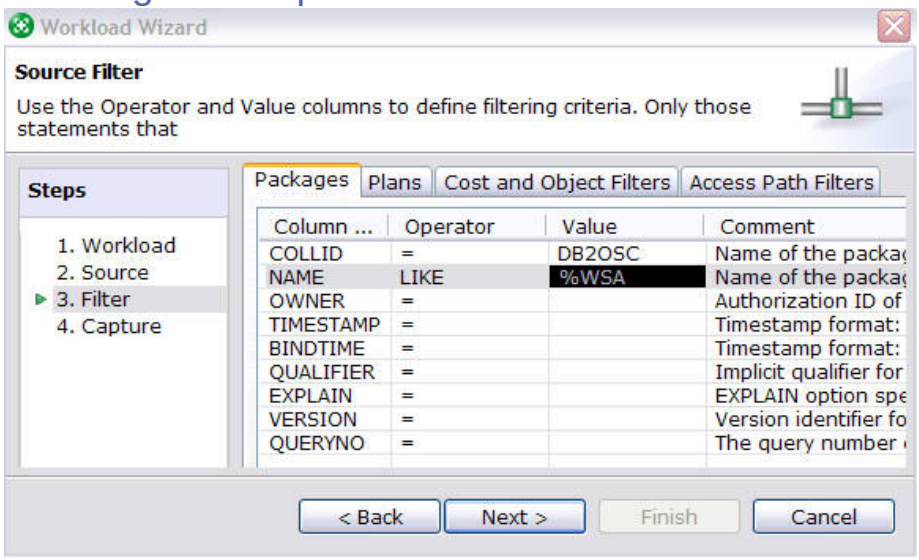
9

Robust filtering options

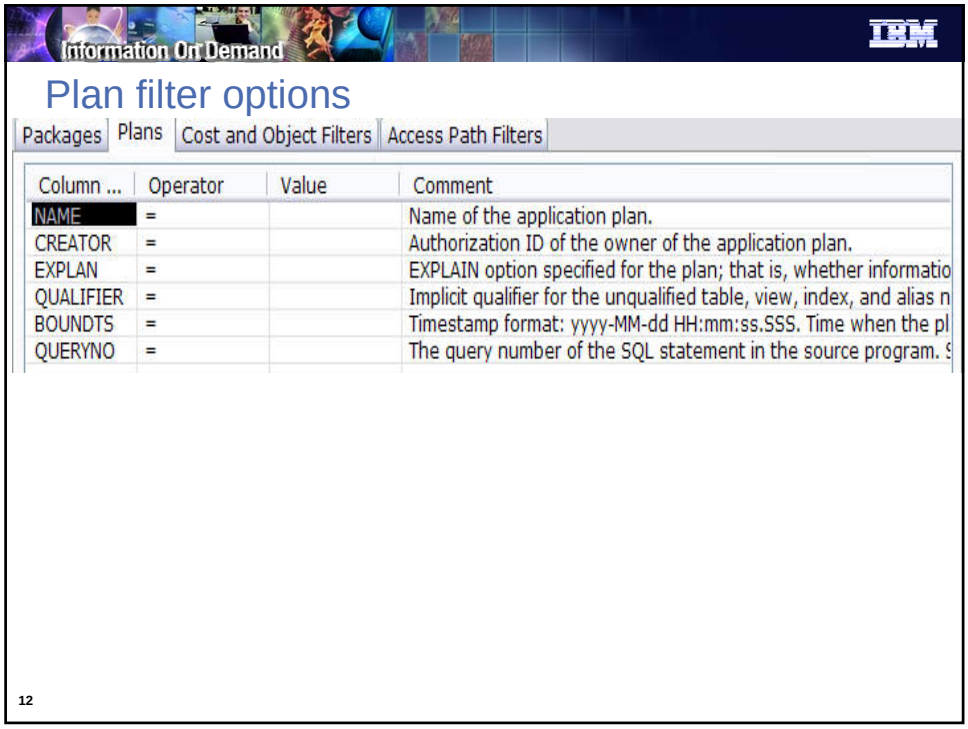
- Package and plan level filters
  - Eg. All SQL with COLLID LIKE ‘OSC%’
- Cost estimate and object filters
  - Eg. Only SQL which access index I1
- Access path filters
  - Eg. Only SQL which perform R-scan…

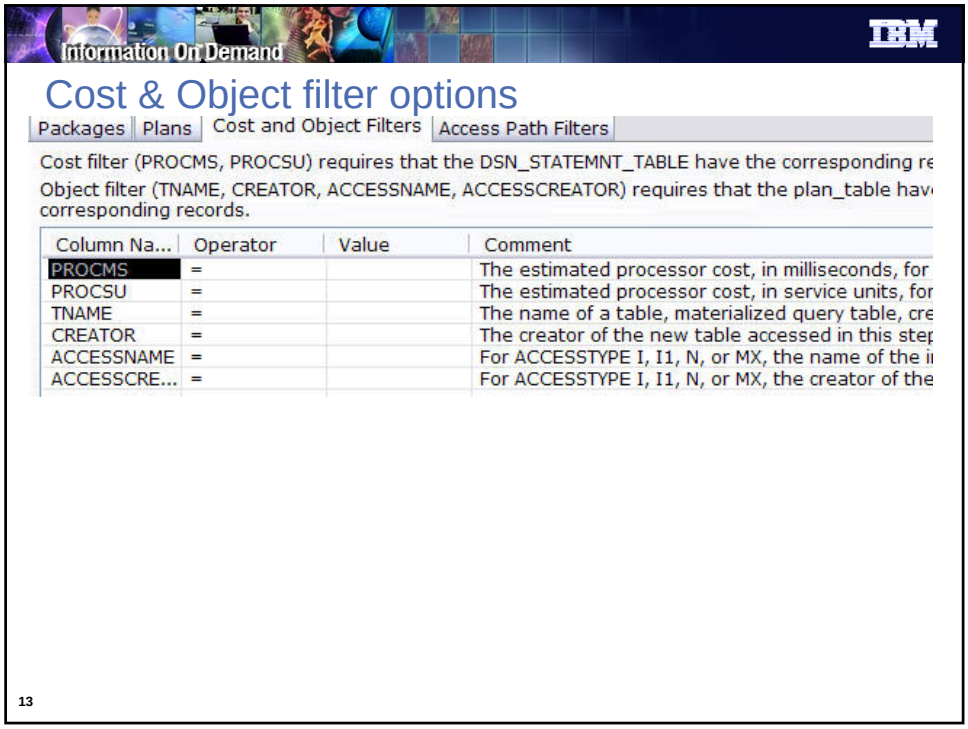
10

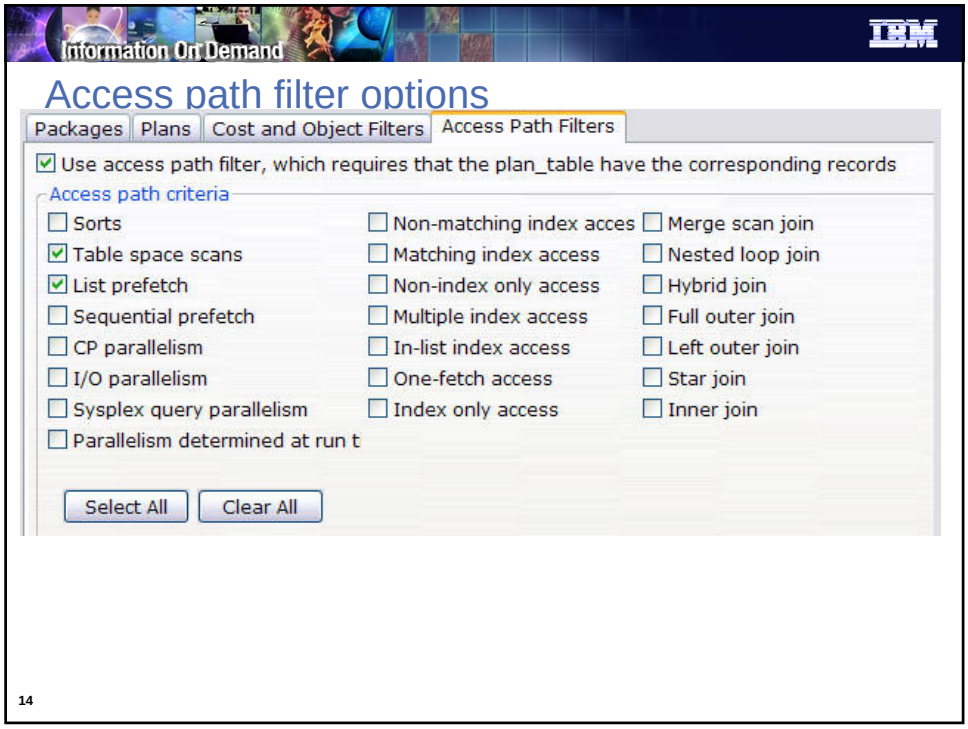
Package filter options

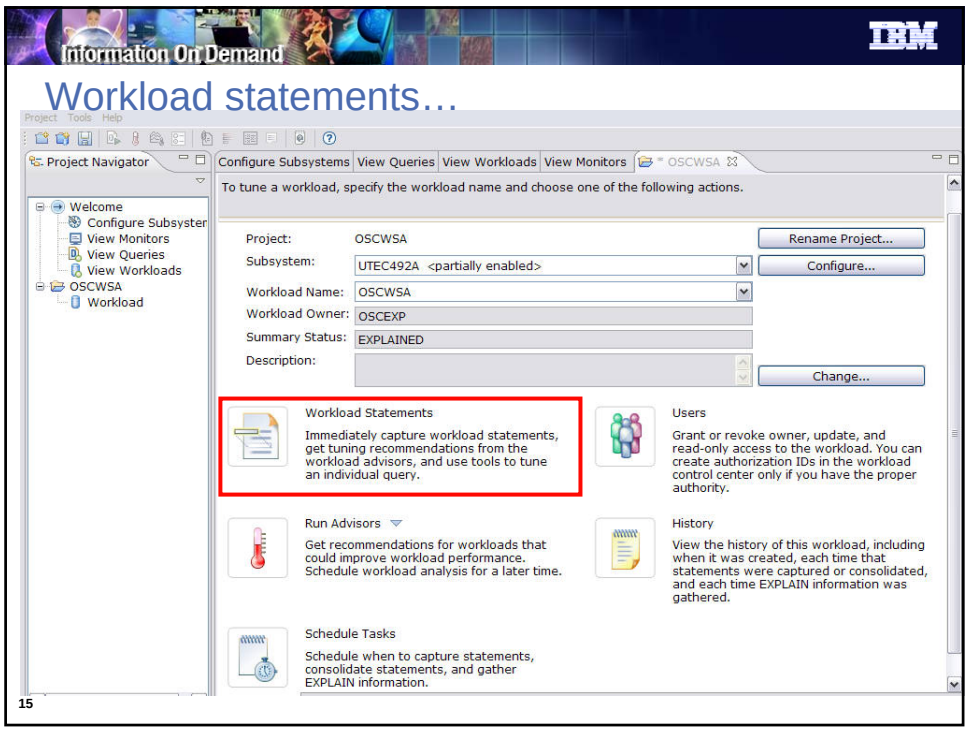


11

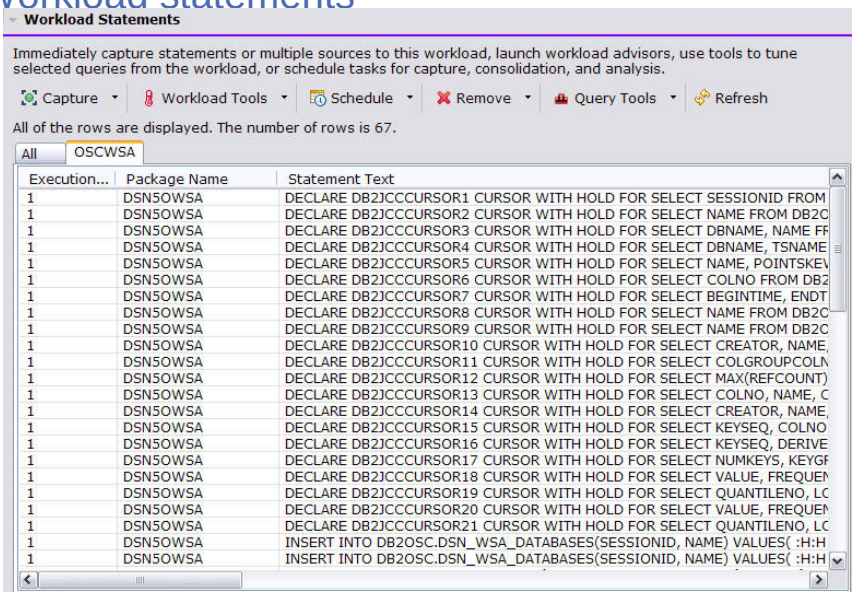

Information On Demand
IBM
Plan filter options
Packages
Plans
Cost and Object Filters
Access Path Filters
Column ...
Operator
Value
Comment
NAME
=
Name of the application plan.
CREATOR
=
Authorization ID of the owner of the application plan.
EXPLAN
=
EXPLAIN option specified for the plan; that is, whether informatio
QUALIFIER
=
Implicit qualifier for the unqualified table, view, index, and alias n
BOUNDTS
=
Timestamp format: yyyy-MM-dd HH:mm:ss.SSS. Time when the pl
QUERYNO
=
The query number of the SQL statement in the source program.
12


Information On Demand
IBM
Cost & Object filter options
Packages
Plans
Cost and Object Filters
Access Path Filters
Cost filter (PROCMS, PROCSU) requires that the DSN_STATEMNT_TABLE have the corresponding re
Object filter (TNAME, CREATOR, ACCESSNAME, ACCESSCREATOR) requires that the plan_table hav
corresponding records.
Column Na...
Operator
Value
Comment
PROCMS
=
The estimated processor cost, in milliseconds, for
PROCSU
=
The estimated processor cost, in service units, for
TNAME
=
The name of a table, materialized query table, cre
CREATOR
=
The creator of the new table accessed in this step
ACCESSNAME
=
For ACCESSTYPE I, I1, N, or MX, the name of the i
ACCESSCRE...
=
For ACCESSTYPE I, I1, N, or MX, the creator of the
13


Information On Demand
IBM
Access path filter options
Packages
Plans
Cost and Object Filters
Access Path Filters
Use access path filter, which requires that the plan_table have the corresponding records
Access path criteria
Sorts
Table space scans
List prefetch
Sequential prefetch
CP parallelism
I/O parallelism
Sysplex query parallelism
Parallelism determined at run t
Non-matching index acces
Matching index access
Non-index only access
Multiple index access
In-list index access
One-fetch access
Index only access
Merge scan join
Nested loop join
Hybrid join
Full outer join
Left outer join
Star join
Inner join
Select All
Clear All
14

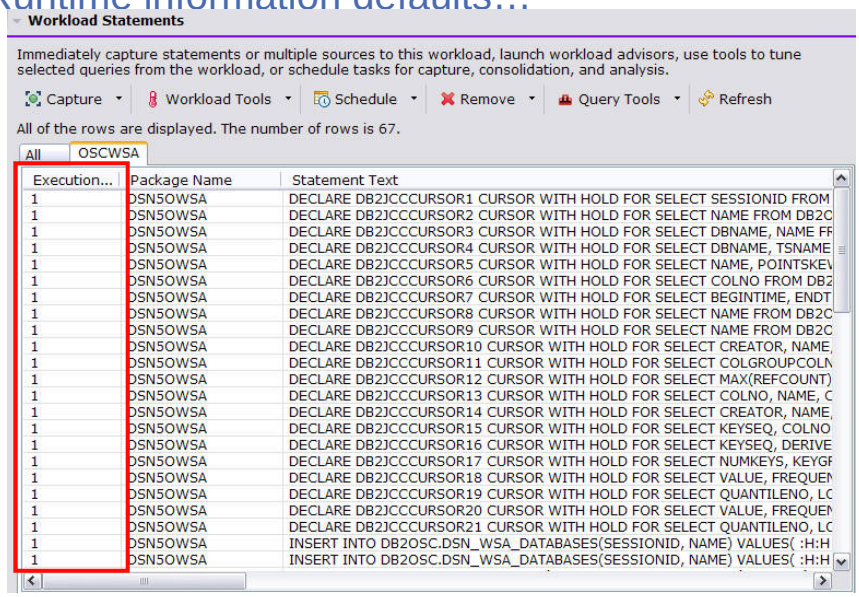

Information On Demand
IBM
Workload statements…
Project Navigator
Welcome
Configure Subsyster
View Monitors
View Queries
View Workloads
OSCWSA
Workload
Configure Subsystems
View Queries
View Workloads
View Monitors
OSCWSA
To tune a workload, specify the workload name and choose one of the following actions.
Project:
OSCWSA
Rename Project...
Subsystem:
UTEC492A <partially enabled>
Configure...
Workload Name:
OSCWSA
Workload Owner:
OSCEXP
Summary Status:
EXPLAINED
Description:
Change...
Workload Statements
Immediately capture workload statements, get tuning recommendations from the workload advisors, and use tools to tune an individual query.
Users
Grant or revoke owner, update, and read-only access to the workload. You can create authorization IDs in the workload control center only if you have the proper authority.
Run Advisors
Get recommendations for workloads that could improve workload performance. Schedule workload analysis for a later time.
History
View the history of this workload, including when it was created, each time that statements were captured or consolidated, and each time EXPLAIN information was gathered.
Schedule Tasks
Schedule when to capture statements, consolidate statements, and gather EXPLAIN information.
15

Workload statements



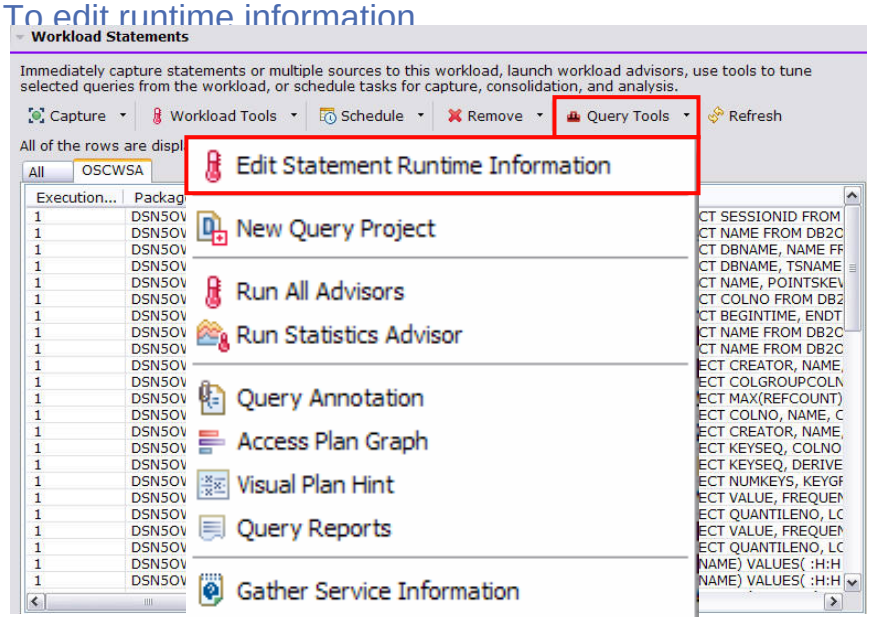
16

Runtime information defaults…



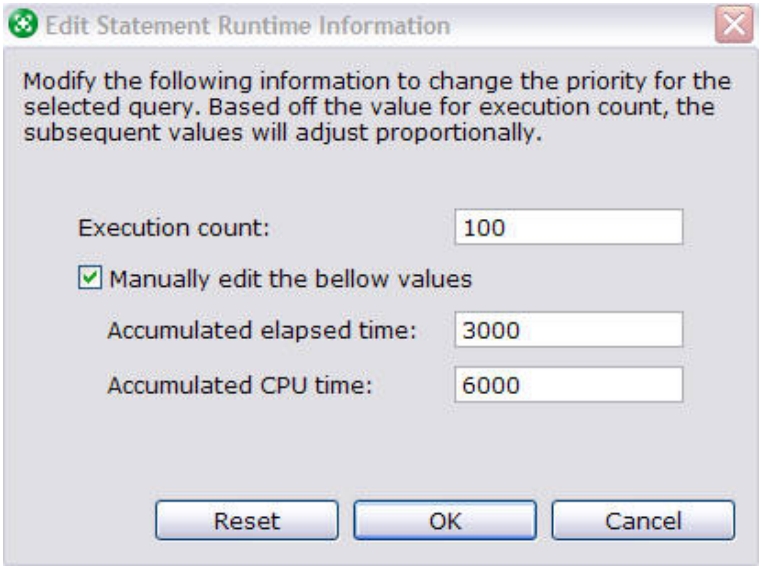
17

To edit runtime information



18

Edit runtime information



19

Implications of no runtime information

- Workload advisors use the runtime information
  - To weight queries (WIA and WSA)
  - To measure performance improvement (WIA)
- Accurate runtime information much more important for WIA than for WSA.
  - Index Advisor uses the frequency of execution, processing cost, and estimated costs as significant factors in determining which indexes provide best value.
  - If you're just running WSA, then runtime information is not as critical for recommendation purposes.

20

Statement cache statements view

Capture | Workload Tools | Schedule | Remove | Query Tools | Refresh

All of the rows are displayed. The number of rows is 22.

All | TUTORIAL_SOURCE

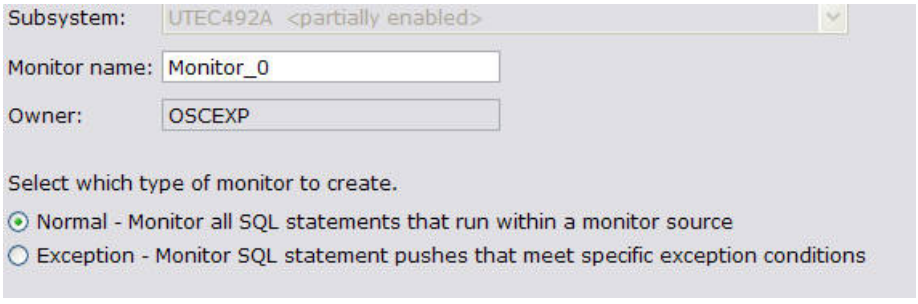
Exec...	Accumulated Elapse...	Accumulated CPU ...	Statement Text
5	0.0024663904	6.994219E-4	SELECT L_ORDERKEY,L_SUPPKEY ,L_SHIPDATE,L_R...
16	0.0014626094	0.0014598203	SELECT count(*) FROM SYSADM.order ,SYSADM.cust...
11	48.690983	0.008925763	SELECT SUM(L_EXTENDEDPRICE*L_DISCOUNT) FRO...
5	20.356037	0.012490061	SELECT * FROM SYSADM.LINEITEM L ,SYSADM.ORDE...
16	70.62658	0.012818922	SELECT SUM(L_EXTENDEDPRICE*L_DISCOUNT) FRO...
16	0.32213017	0.019210016	SELECT C_NATIONKEY, SUM(C_ACCTBAL) FROM SYS...
11	127.6105	0.021940278	SELECT L_RETURNFLAG, L_LINESTATUS , SUM(L_QU...
16	0.21988066	0.036937166	SELECT C_CUSTKEY, C_NAME , SUM(L_EXTENDEDPR...
16	11.466922	0.040909052	SELECT L_DISCOUNT , SUM(L_QUANTITY) , AVG(L_Q...
9	43.5331	0.044824567	SELECT L_QUANTITY, COUNT(*) ,MIN(L_TAX) ,MAX(L...
11	26.944782	0.04755859	SELECT L_QUANTITY, COUNT(*),MIN(L_TAX) ,MAX(L...
9	6.865624	0.11236204	SELECT L_LINESTATUS, COUNT(*) FROM SYSADM.LI...
5	433.53644	0.3007743	SELECT O_ORDERPRIORITY, COUNT(*) FROM SYSAD...
5	27.1646	0.35811093	SELECT L_SUPPKEY, COUNT(*),MIN(L_TAX) ,MAX(L_...
5	122.8509	2.2276213	SELECT DISTINCT O_ORDERKEY FROM SYSADM.LINE...
5	38.991688	2.375377	SELECT S_SUPPKEY, S_NAME , SUM(L_EXTENDEDPR...
11	30.397251	2.3755095	SELECT P_PARTKEY,P_NAME,P_BRAND FROM SYSAD...
5	47.044178	4.1442504	SELECT L_ORDERKEY, SUM(L_EXTENDEDPRICE) AS ...
13	285.1192	4.574241	SELECT O_ORDERPRIORITY, COUNT(*) FROM SYSAD...
11	165.84152	4.8676553	SELECT DISTINCT O_ORDERKEY FROM SYSADM.LINE...
13	150.82983	7.8861747	SELECT L_SUPPKEY,L_SHIPINSTRUCT,L_SHIPMODE,...
11	118.390465	11.202963	SELECT C_CUSTKEY, C_NAME , SUM(L_EXTENDEDPR...

21

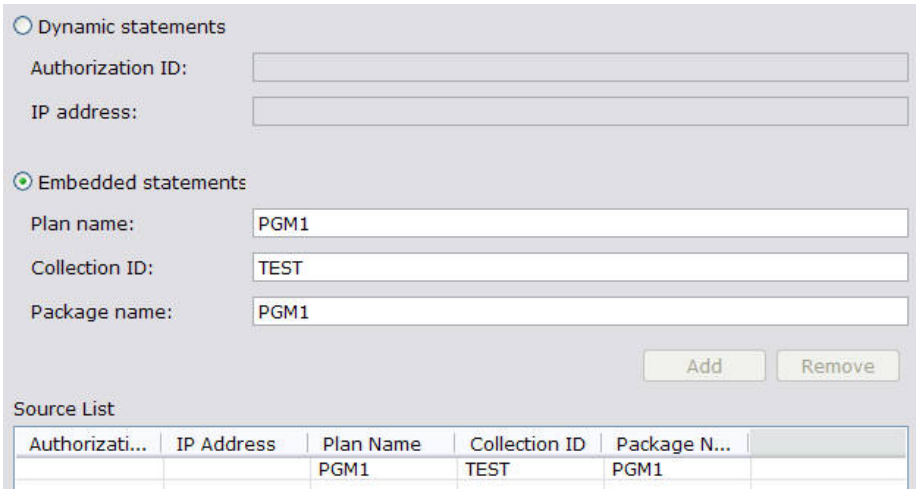
Monitor workload

- Normal monitor workload
  - All SQL based on certain criteria
  - Obtain runtime information for all monitored SQL (including static)
  - Useful to profile an application
    - Which SQL running most frequently?
    - Which queries consuming most elapsed / execution time?
  - Useful to identify which SQL are consuming most resources, tuning opportunities
- Exception monitor
  - Workload generated based on exception criteria
    - All SQL for package > x seconds elapsed
    - All SQL for package > x seconds elapsed

Create normal monitor

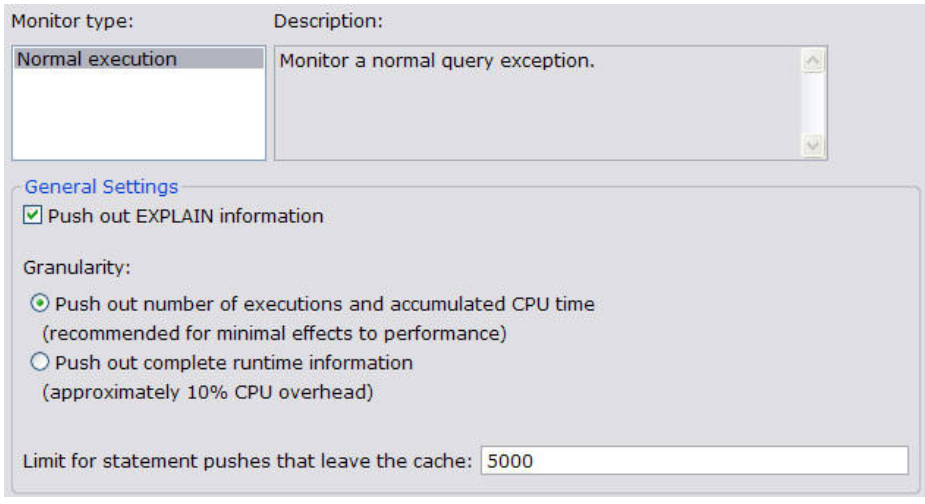


Add monitor sources



24

Set monitor options



25

Monitor statement view

All of the rows are displayed. The number of rows is 3.

All | Monitor

Exec...	Accumulated CPU...	Accumulated El...	Statement Text
20	57.242344	59.397785	DECLARE C2 C...
55	0.0119307535	0.013347485	DECLARE C2 C...
13	0.003889464	0.005787883	DECLARE C1 C...

26

Exception monitor options

Monitor type: Description:

- [x] CPU time exception
- [x] Relative CPU time exce

Monitor Settings

Specify the type of normal monitor information to push out, or specify threshold criteria for the selected exception monitor types.

CPU time threshold: seconds

Relative CPU time threshold: %

27

Run advisors on the workload

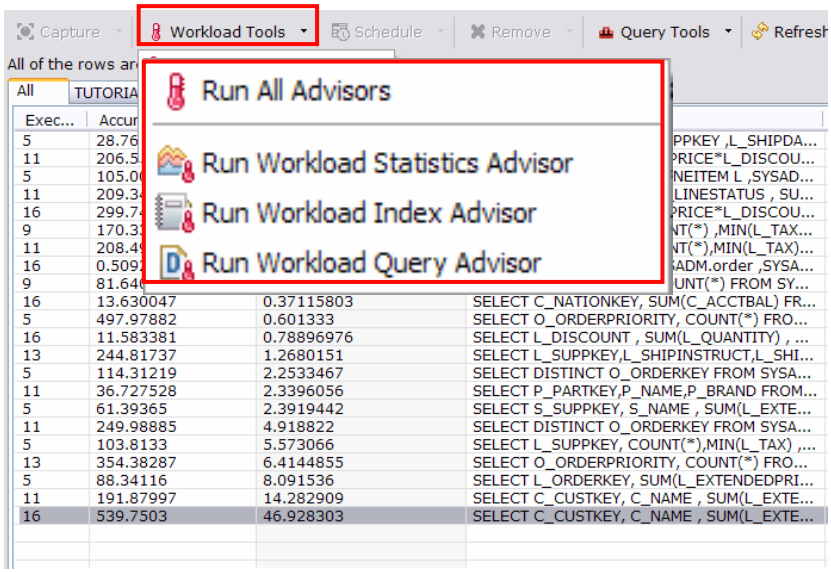
28

Workload advisors

- Statistics advisor
  - Generate statistics recommendations based on SQL within the workload
- Index advisor
  - Generate index recommendations based on SQL within the workload
- Query advisor
  - Analyze workload SQL, provide query rewrite suggestions

29

Workload advisors…



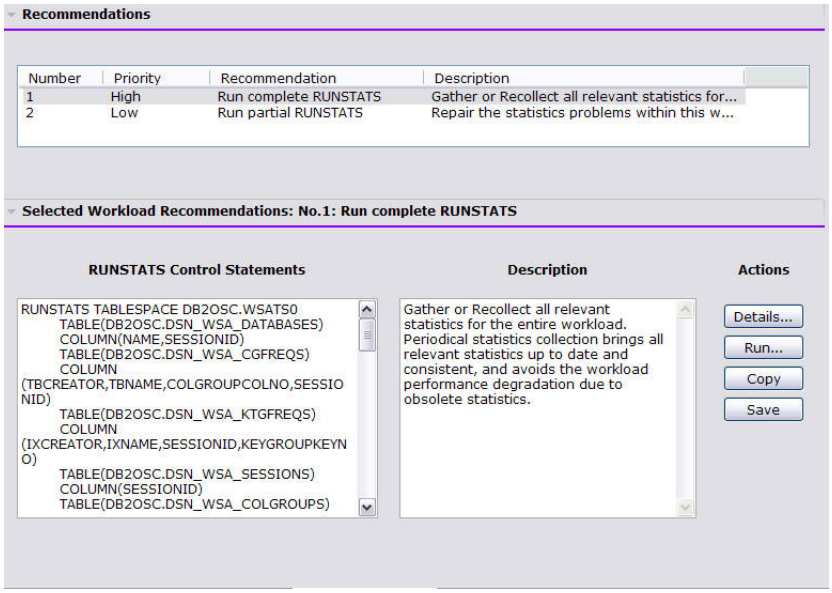
30

Why statistics advisor?

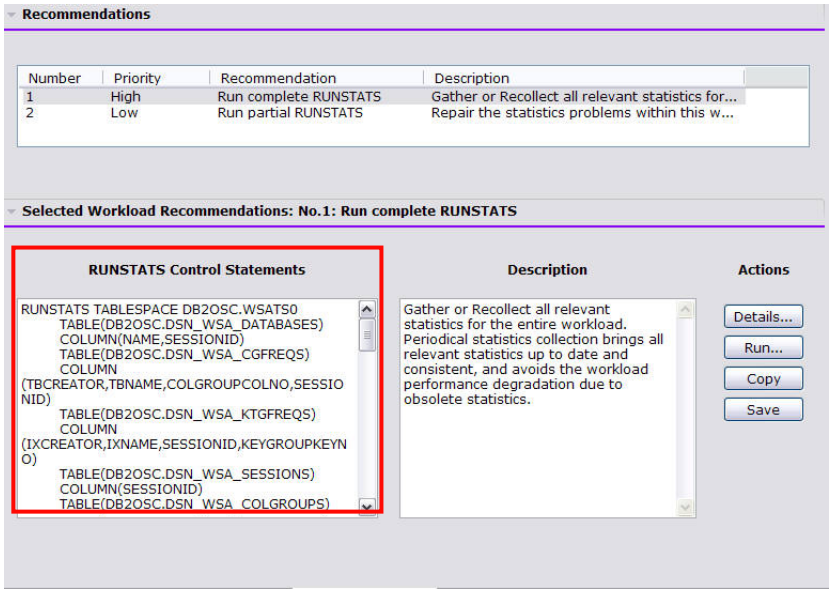
- Statistics are collected to help optimizer in costing available access paths
  - Say you interview for a DBA position with three companies, and they all provided offers.
    - What is the salary?
    - How much vacation?
    - How is the commute?
    - Will you have to work weekends, nightshift, how often?
  - What if you had to make a decision between the three offers, and you did not have (accurate) answers to the above questions?
  - Without adequate answers to these questions, what are the odds you'd be satisfied with your choice?
- Statistics advisor is asking these questions of your data for your queries.

31

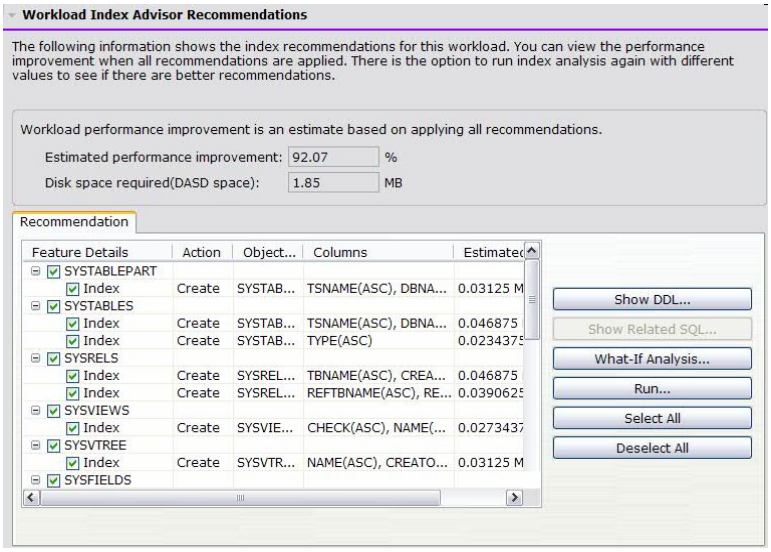
Statistics advisor



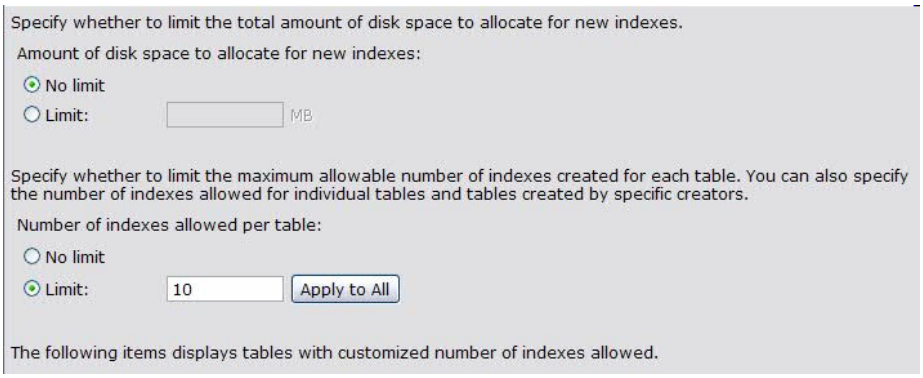
RUNSTATS targeted to your queries



Index advisor recommendations



What-if analysis options



Query advisor summary

Workload Query Advisor Recommendations Summary

The following is a summary of the queries analyzed in the workload. Use this criteria to filter the view for specific statements.

Statements Sorted by	Number
Statements Analyzed Successfully	22
Statements with Warnings	3
Number of High Severity Warnings	0
Number of Medium Severity Warnings	0
Number of Low Severity Warnings	6
Statements with High Severity Warnings	0
Statements with Medium Severity Warnings	0
Statements with Low Severity Warnings	3

View statements that meet the following criteria:

Degree of warning severity:
- ☑ High severity
- ☑ Medium severity
- ☑ Low severity
- ☑ Show statements that do not contain warnings

Restore Defaults | Save as Defaults

36

Individual SQL recommendation (in-list PTC)

SQL Text

```
   , SYSADM.LINEITEM AS L
   , SYSADM.ORDER AS O
WHERE ( L.L_RECEIPTDATE <= '1999-12-31'
      AND L.L_SHIPDATE BETWEEN '1998-01-01' AND '1998-12-31'
      AND L.L_RETURNFLAG IN ( 'A', 'R' )
      AND L.L_SHIPMODE IN ( 'SHIP', 'AIR', 'RAIL', 'TRUCK' )
      AND S.S_SUPPKEY IN ( 1, 22, 333, 4444, 55555 )
```

Selected Recommendation:

Description	Explanation
Consider copying the following predicate on column S_SUPPKEY in table SYSADM.SUPPLIER to column L_SUPPKEY in table SYSADM.LINEITEM: S.S_SUPPKEY IN (1 , 22 , 333 , 4444 , 55555) which might filter the table earlier. Check the explanation for this warning for more details about possible impact and examples.	Adding local transitive closure predicates might improve the performance of the specified query. Predicate transitive closure is the process whereby DB2 copies a local predicate from one table to another before joining the tables. For example, consider the following predicate: WHERE T1.C1 = T2.C1 AND T1.C1 = X DB2 can copy the local predicate T1.C1 = X to T2.C1.

37

Workload summary

- OSC and OE provide robust workload capabilities
  - Multiple sources supported
  - Ability to profile an application
  - Generate targeted statistics
  - Targeted index recommendations
  - Workload level query analysis
- After performing workload level tuning, use individual query level tuning for remaining top cpu consuming SQL

Individual query analysis and and tuning

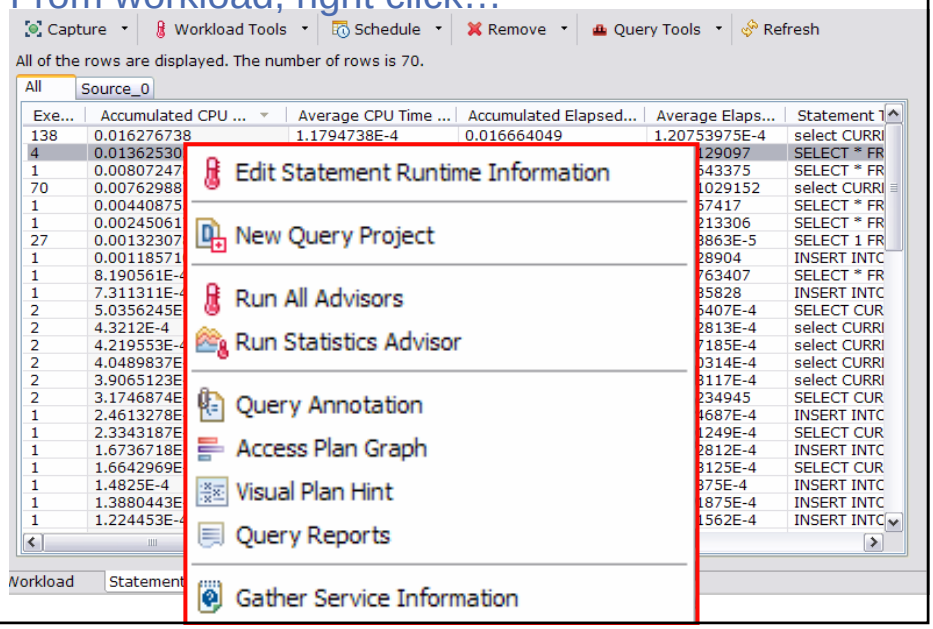
Individual query analysis and tuning

- Query tuning
  - Query Advisor
  - Statistics Advisor
  - Access Path Advisor
  - Query annotation
  - Query report
  - Access plan graph
  - Visual plan hint

Tuning a query

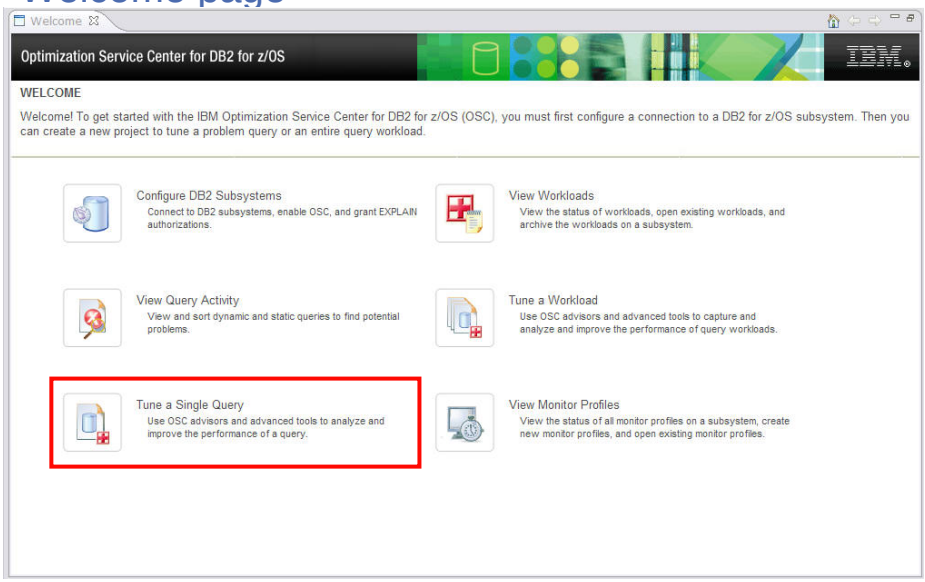
- Steps to tuning a query
  - Identify statement
- Collect information regarding SQL
  - Formatted SQL
  - Physical design of referenced objects
  - Statistics
  - Access path
  - Advisor output (like having another DBA perspective)
  - Mentally compute, is access path efficient?
  - Could we do better?

From workload, right click…



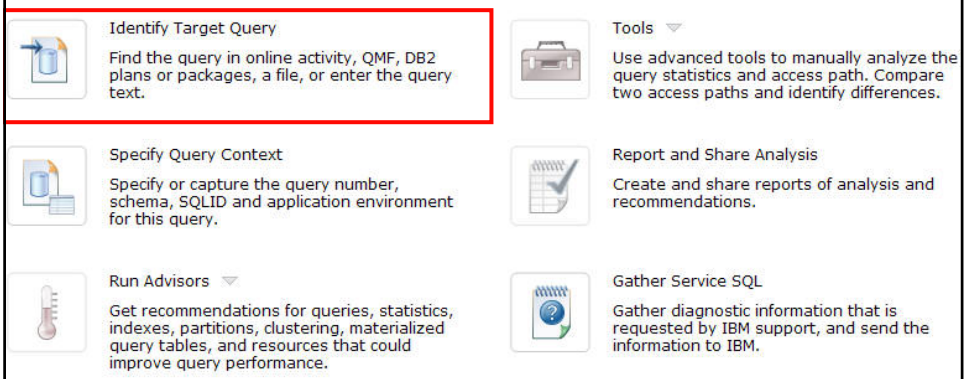
42

Welcome page



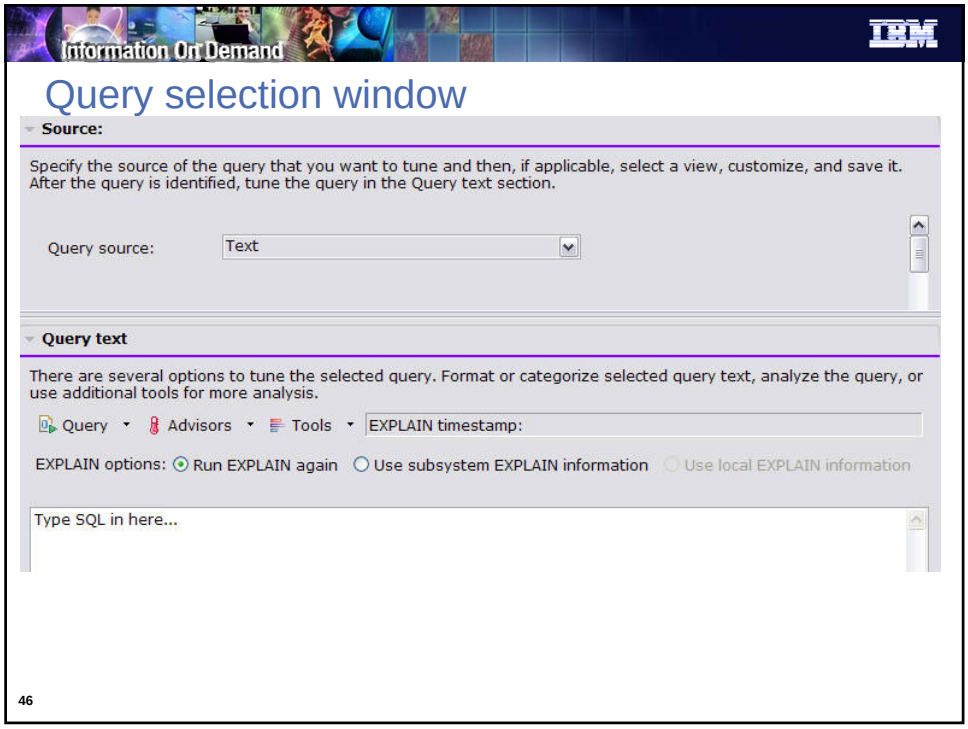
43

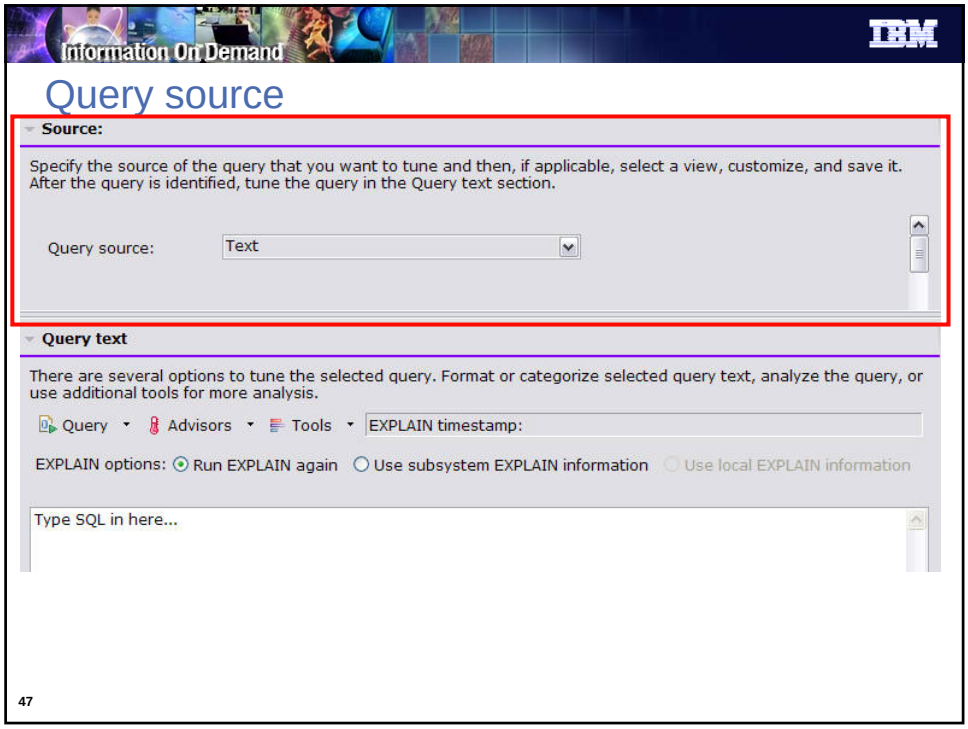
Choose identify target query…



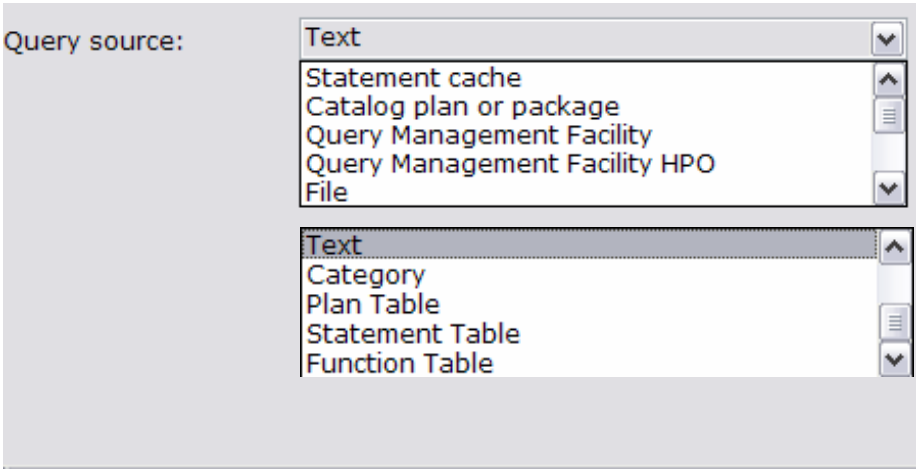
Query options

- Top of window, select query source
  - Top of window varies based on selected source
- Bottom of window
  - Show selected SQL
  - Tuning options


Information On Demand
IBM
Query selection window
Source:
Specify the source of the query that you want to tune and then, if applicable, select a view, customize, and save it. After the query is identified, tune the query in the Query text section.
Query source:
Text
Query text
There are several options to tune the selected query. Format or categorize selected query text, analyze the query, or use additional tools for more analysis.
Query
Advisors
Tools
EXPLAIN timestamp:
EXPLAIN options:
Run EXPLAIN again
Use subsystem EXPLAIN information
Use local EXPLAIN information
Type SQL in here...
46

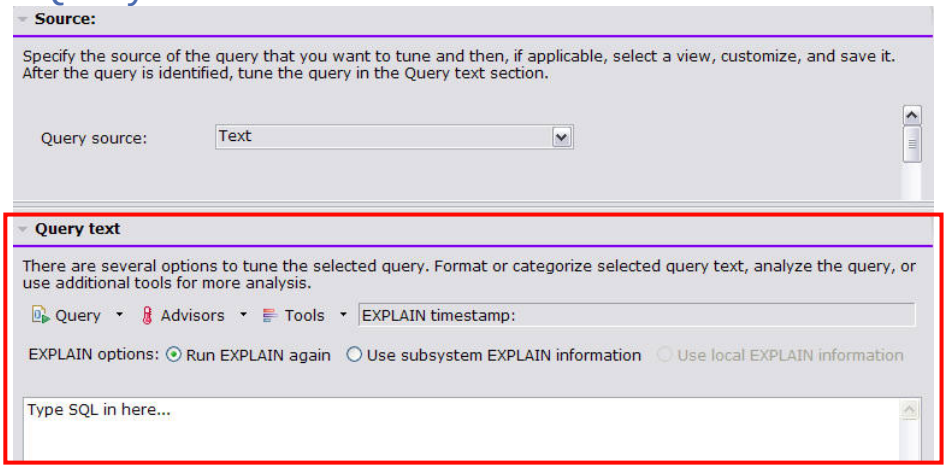

Information On Demand
IBM
Query source
Source:
Specify the source of the query that you want to tune and then, if applicable, select a view, customize, and save it. After the query is identified, tune the query in the Query text section.
Query source:
Text
Query text
There are several options to tune the selected query. Format or categorize selected query text, analyze the query, or use additional tools for more analysis.
Query
Advisors
Tools
EXPLAIN timestamp:
EXPLAIN options:
Run EXPLAIN again
Use subsystem EXPLAIN information
Use local EXPLAIN information
Type SQL in here...
47

Choose query source…



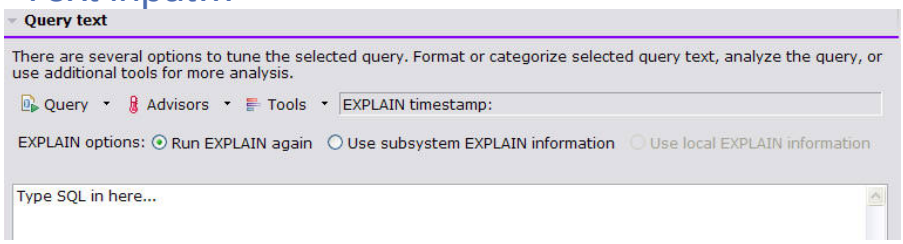
48

Query text



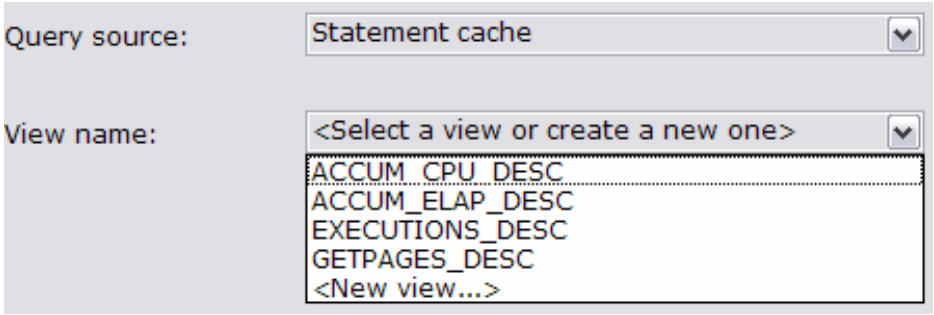
49

Text input…



50

Source: Statement cache…



- Improved statement cache interface!
  - Use predefined cache views sorted in popular use sequence
  - Define and save your own display of the cache

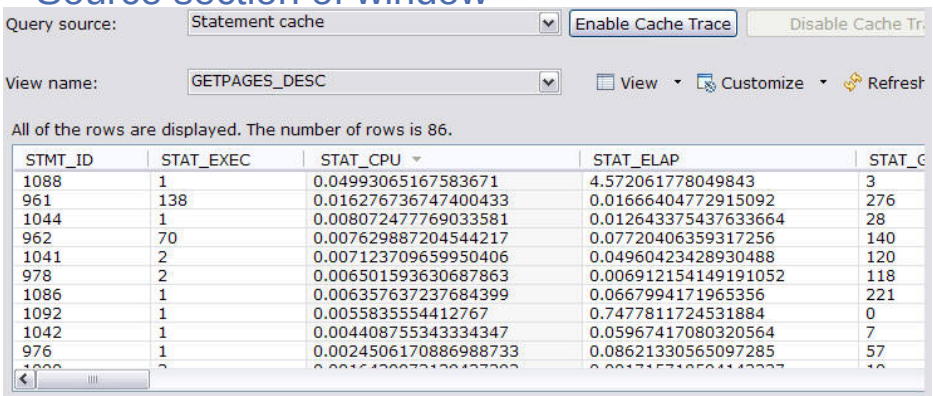
51

Other cache improvements

- Design your own cache display view
  - Choose filter criteria
  - Choose how to sort rows retrieved from cache
  - Choose which columns to display
  - After rows retrieved, you can re-sort the rows displayed on the screen also
- Vastly improved statement cache interface
  - One-time setup, repeating use

52

Source section of window

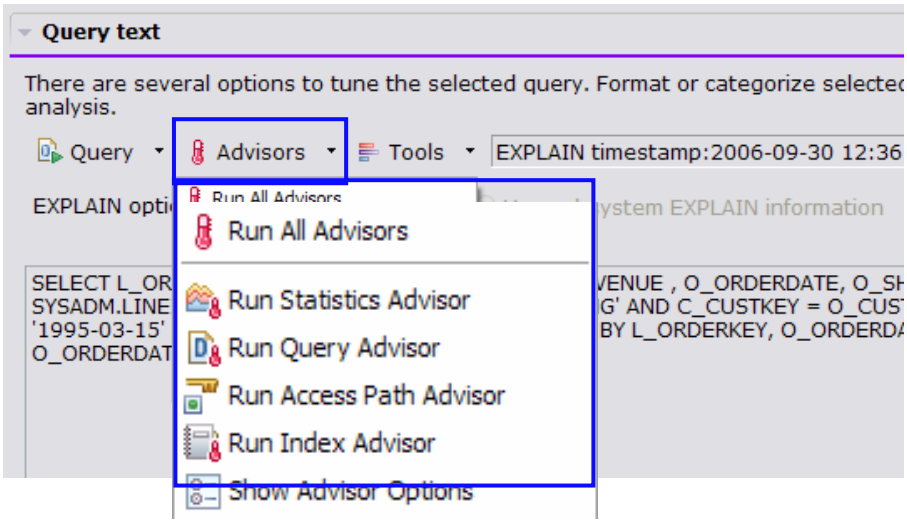


STMT_ID	STAT_EXEC	STAT_CPU	STAT_ELAP	STAT_G
1088	1	0.04993065167583671	4.572061778049843	3
961	138	0.016276736747400433	0.01666404772915092	276
1044	1	0.008072477769033581	0.012643375437633664	28
962	70	0.007629887204544217	0.07720406359317256	140
1041	2	0.007123709659950406	0.04960423428930488	120
978	2	0.006501593630687863	0.006912154149191052	118
1086	1	0.006357637237684399	0.0667994171965356	221
1092	1	0.0055835554412767	0.7477811724531884	0
1042	1	0.004408755343334347	0.05967417080320564	7
976	1	0.0024506170886988733	0.08621330565097285	57

- Double left click to see SQL text
- Left click on row to select SQL
- Left-click on column header to sort retrieved records by column

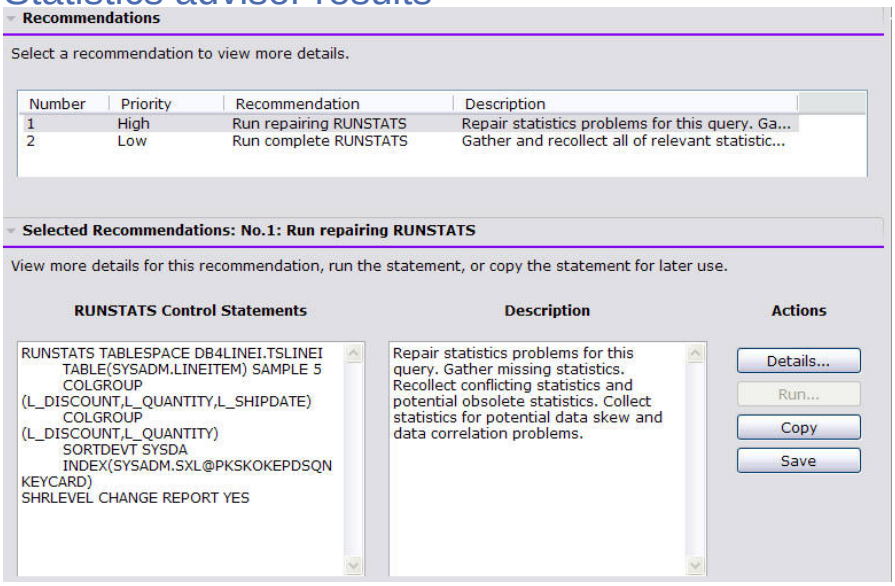
53

Query advisors



54

Statistics advisor results



55

Query advisor results

SQL Text

```
    , SYSADM.LINEITEM AS L
    , SYSADM.ORDER AS O
WHERE ( L.L_RECEIPTDATE <= '1999-12-31'
      AND L.L_SHIPDATE BETWEEN '1998-01-01' AND '1998-12-31'
      AND L.L_RETURNFLAG IN ( 'A', 'R' )
      AND L.L_SHIPMODE IN ( 'SHIP', 'AIR', 'RAIL', 'TRUCK' )
      AND S.S_SUPPKEY IN ( 1, 22, 333, 4444, 55555 )
```

Selected Recommendation:

Description	Explanation
Consider copying the following predicate on column S_SUPPKEY in table SYSADM.SUPPLIER to column L_SUPPKEY in table SYSADM.LINEITEM: S.S_SUPPKEY IN (1 , 22 , 333 , 4444 , 55555) which might filter the table earlier. Check the explanation for this warning for more details about possible impact and examples.	Adding local transitive closure predicates might improve the performance of the specified query. Predicate transitive closure is the process whereby DB2 copies a local predicate from one table to another before joining the tables. For example, consider the following predicate: WHERE T1.C1 = T2.C1 AND T1.C1 = X DB2 can copy the local predicate T1.C1 = X to T2.C1.

56

Access path advisor

Access Path Warning List

The following list shows potential problems in the selected statement's access path. Select a warning to view a more detailed description. Make the necessary changes to avoid this warning in the future.

Severity	Query Block Number	Plan Number	Description
APA_HIGH_SEVERITY	1	2	The inner table SYSIBM.SYS...

Access Path Warning Details

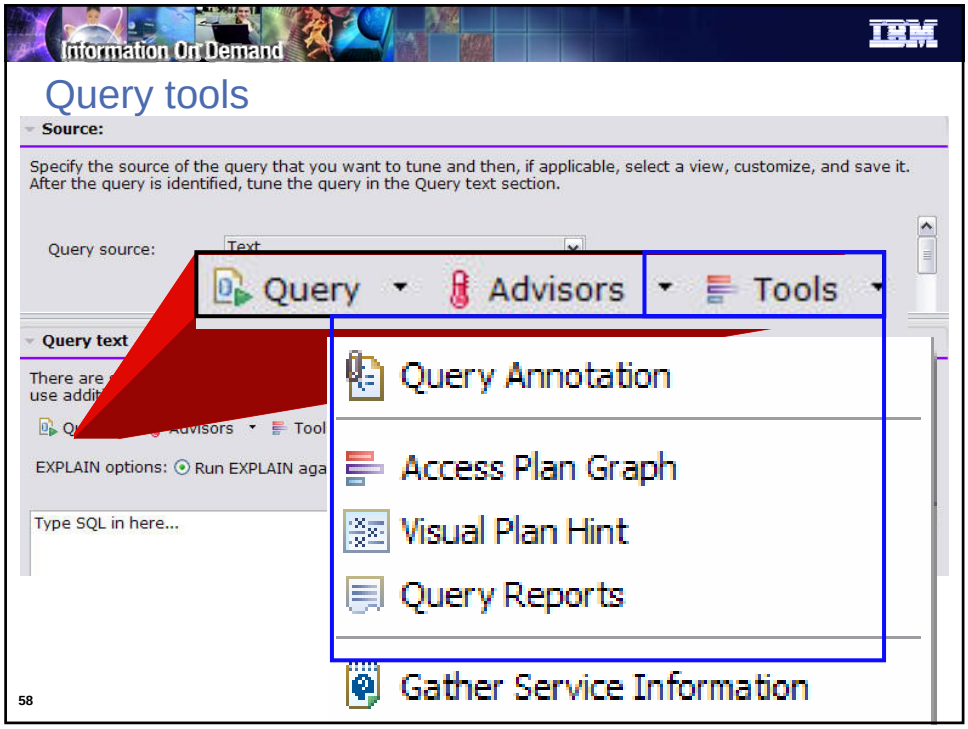
Description	Explanation
The inner table SYSIBM.SYSTABLESPACESTATS in the nested loop join is accessed by a relational scan. When a large number of records are returned after the outer table is accessed, DB2 might be using a inefficient access path. Check the explanation for this	When the DB2 optimizer chooses a nested loop join, DB2 first scans the outer table and then scans the inner table one time for each qualifying row in the outer table. The DB2 optimizer might choose to access the inner table by using a table space scan.

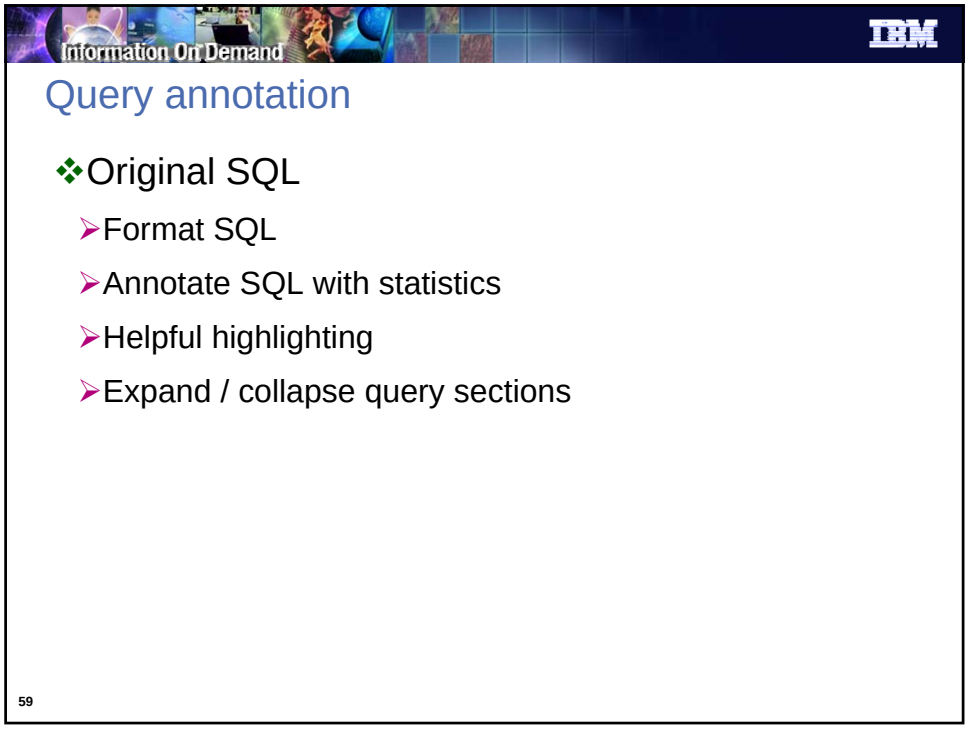
PLAN_TABLE record

The following row in the plan table related with this warning.

QBLO...	PLANNO	MIXOP...	METHOD	CREAT...	TNAME	CORRE...	ACCES...	ACCES...	ACCES...	MATCH..
1	2	0	1	SYSIBM	SYSTAB...	T	R			0
1	1	0	0	SYSIBM	SYSTAB...	P	I	SYSIBM	DSNDP...	0

57


Information On Demand
IBM
Query tools
Source:
Specify the source of the query that you want to tune and then, if applicable, select a view, customize, and save it.
After the query is identified, tune the query in the Query text section.
Query source:
Text
Query
Advisors
Tools
Query text
There are
use addi
Advisors
Tool
EXPLAIN options:
Run EXPLAIN aga
Type SQL in here...
Query Annotation
Access Plan Graph
Visual Plan Hint
Query Reports
Gather Service Information
58


Information On Demand
IBM
Query annotation
Original SQL
Format SQL
Annotate SQL with statistics
Helpful highlighting
Expand / collapse query sections
59

Original SQL

```
SELECT S_SUPPKEY, S_NAME ,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS
  REVENUE FROM SYSADM.ORDER , SYSADM.LINEITEM
  , SYSADM.SUPPLIER WHERE L_ORDERKEY =
  O_ORDERKEY AND S_SUPPKEY = L_SUPPKEY AND
  O_ORDERDATE >= DATE('1993-10-01') AND
  O_ORDERDATE < DATE('1993-10-01') + 3 MONTH AND
  L_RETURNFLAG = 'R' AND S_NATIONKEY IN (1,2,3,4,5)
  AND S_SUPPKEY < 10000 GROUP BY S_SUPPKEY,
  S_NAME HAVING SUM(L_EXTENDEDPRICE*(1-
  L_DISCOUNT)) > 1000000 ORDER BY REVENUE DESC
  QUERYNO 3
```


Formatted

```
SELECT SYSADM.SUPPLIER.S_SUPPKEY
     , SYSADM.SUPPLIER.S_NAME
     , SUM( SYSADM.LINEITEM.L_EXTENDEDPRICE * ( 1 - SYSADM.LINEITEM.L_DISCOUNT ) ) AS REVENUE
FROM SYSADM.SUPPLIER
   , SYSADM.ORDER
   , SYSADM.LINEITEM
WHERE ( SYSADM.LINEITEM.L_ORDERKEY = SYSADM.ORDER.O_ORDERKEY
      AND SYSADM.SUPPLIER.S_SUPPKEY = SYSADM.LINEITEM.L_SUPPKEY
      AND SYSADM.LINEITEM.L_RETURNFLAG = 'R'
      AND SYSADM.ORDER.O_ORDERDATE < ( DATE( '1993-10-01' ) + 3 MONTHS )
      AND SYSADM.ORDER.O_ORDERDATE >= DATE( '1993-10-01' )
      AND SYSADM.SUPPLIER.S_NATIONKEY IN ( 1, 2, 3, 4, 5 )
      AND SYSADM.SUPPLIER.S_SUPPKEY < 10000
    )
GROUP BY SYSADM.SUPPLIER.S_SUPPKEY
       , SYSADM.SUPPLIER.S_NAME
HAVING SUM( SYSADM.LINEITEM.L_EXTENDEDPRICE * ( 1 - SYSADM.LINEITEM.L_DISCOUNT ) ) > 1000000
ORDER BY REVENUE DESC
```

Annotated

```
SELECT SYSADM.SUPPLIER.S_SUPPKEY
     , SYSADM.SUPPLIER.S_NAME
     , SUM( SYSADM.LINEITEM.L_EXTENDEDPRICE * ( 1 - SYSADM.LINEITEM.L_DISCOUNT ) ) AS REVENUE
FROM SYSADM.SUPPLIER                                          CARDF=10,000      QUALIFIED_ROWS=1,526.2466  NPAGESF=402
    , SYSADM.ORDER                                            CARDF=1,500,000 QUALIFIED_ROWS=57,114.59  NPAGESF=42,417
    , SYSADM.LINEITEM                                         CARDF=4,254,339 QUALIFIED_ROWS=1,048,737.0 NPAGESF=141,743
WHERE ( SYSADM.LINEITEM.L_ORDERKEY = SYSADM.ORDER.O_ORDERKEY          COLCARDF=1,063,455/1,500,00
                                                                       MAX_FREQ=0.0001645%/0.000066666666
                                                                       FF=6.666666649834951E-7
      AND SYSADM.SUPPLIER.S_SUPPKEY = SYSADM.LINEITEM.L_SUPPKEY        COLCARDF=10,000/10,000
                                                                       MAX_FREQ=99.99%/0.011917244958617542%
                                                                       FF=9.99999901978299E-5
      AND SYSADM.LINEITEM.L_RETURNFLAG = 'R'                           COLCARDF=4
                                                                       MAX_FREQ=50.75460606218732%
                                                                       FF=0.24650996923446655
      AND SYSADM.ORDER.O_ORDERDATE < ( DATE( '1993-10-01' ) + 3 MONTHS )   COLCARDF=2,304
                                                                       MAX_FREQ=0.0414%
                                                                        FF=0.30288922786712646
                                                                        LOW2KEY=1992-01-02 HIGH2KEY=1998-08-01
      AND SYSADM.ORDER.O_ORDERDATE >= DATE( '1993-10-01' )             COLCARDF=2,304
                                                                        MAX_FREQ=0.0414%
                                                                        FF=0.7351873517036438
                                                                       LOW2KEY=1992-01-02 HIGH2KEY=1998-08-01
      AND SYSADM.SUPPLIER.S_NATIONKEY IN ( 1, 2, 3, 4, 5 )                      COLCARDF=25
                                                                        MAX_FREQ=4.21%
                                                                       FF=0.1526399850845337
      AND SYSADM.SUPPLIER.S_SUPPKEY < 10000                                     COLCARDF=10,000
                                                                       MAX_FREQ=99.99%
                                                                       FF=0.9998999834060669
                                                                       LOW2KEY=2
                                                                       HIGH2KEY=9999
    )
GROUP BY SYSADM.SUPPLIER.S_SUPPKEY
      , SYSADM.SUPPLIER.S_NAME
HAVING SUM( SYSADM.LINEITEM.L_EXTENDEDPRICE * ( 1 - SYSADM.LINEITEM.L_DISCOUNT ) ) > 1000000      FF=0.3333333134651184
ORDER BY REVENUE DESC
```

62

Annotated within OSC

Formatted Query	Annotation
⊟ SELECT SYSADM.SUPPLIER.S_SUPPKEY	
, SYSADM.SUPPLIER.S_NAME	
, SUM(SYSADM.LINEITEM.L_EXTENDEDPRICE * (1 - SYSADM.LINEITEM.L_DI	
FROM SYSADM.SUPPLIER	CARDF=10,000 QUALIFIED_ROWS=1,526.2466 NPAGES
, SYSADM.ORDER	CARDF=1,500,000 QUALIFIED_ROWS=57,114.59 NPAGE
, SYSADM.LINEITEM	CARDF=4,254,339 QUALIFIED_ROWS=1,048,737.0 NPAGE
WHERE (SYSADM.LINEITEM.L_ORDERKEY = SYSADM.ORDER.O_ORDERKEY	COLCARDF=1,063,455/1,500,000 MAX_FREQ=1.6453554
AND SYSADM.SUPPLIER.S_SUPPKEY = SYSADM.LINEITEM.L_SUPPKEY	COLCARDF=10,000/10,000 MAX_FREQ=99.99%/0.011
AND SYSADM.LINEITEM.L_RETURNFLAG = 'R'	COLCARDF=4 MAX_FREQ=50.754606062187
AND SYSADM.ORDER.O_ORDERDATE < (DATE('1993-10-01') + 3 MONTHS	COLCARDF=2,304 MAX_FREQ=0.0414%
AND SYSADM.ORDER.O_ORDERDATE >= DATE('1993-10-01')	COLCARDF=2,304 MAX_FREQ=0.0414%
AND SYSADM.SUPPLIER.S_NATIONKEY IN (1, 2, 3, 4, 5)	COLCARDF=25 MAX_FREQ=4.21%
AND SYSADM.SUPPLIER.S_SUPPKEY < 10000	COLCARDF=10,000 MAX_FREQ=99.99%
)	
GROUP BY SYSADM.SUPPLIER.S_SUPPKEY	
, SYSADM.SUPPLIER.S_NAME	
HAVING SUM(SYSADM.LINEITEM.L_EXTENDEDPRICE * (1 - SYSADM.LINEITEM.L_	FF=0
ORDER BY REVENUE DESC	

63

Annotation table details

- Table annotations
  - CARDF – number of rows in the table
  - NPAGESF – number of pages
  - Qualified rows – optimizer estimated number of rows after local filtering (as though table is the outer table)
- Value
  - Poor join sequence often due to inaccurate filtering estimate at the single table level
  - Qualified rows shows optimizer estimate, is it accurate?
  - CARDF, NPAGESF – useful to have this embedded as you're looking at SQL and predicates

Annotation table close-ups

```
SELECT  cols…
FROM SYSADM.SUPPLIER
                        CARDF=10,000
                        QUAL_ROWS=1,526.2466
                        NPAGESF=402
      , SYSADM.ORDER
                        CARDF=1,500,000
                        QUAL_ROWS=57,114.59
                        NPAGESF=42,417
      , SYSADM.LINEITEM
                        CARDF=4,254,339
                        QUAL_ROWS=1,048,737.0
                        NPAGESF=141,743
```

Annotation predicate details

- Column information
  - COLCARDF – number of distinct values for a column. For join predicate, both COLCARDF's shown
    - Show selectivity of each predicate
  - Max Freq – Maximum frequency (if collected)
    - Skew statistics collected?
    - Show most skewed frequency (indicator of degree of skew)

66

Annotation predicate details

- Column information (cont.)
  - FF (filter factor) – optimizer single predicate filter factor estimate
    - Show optimizer filtering estimate
    - Is it accurate? (Compare to actual filtering)
  - LOW2KEY / HIGH2KEY – shown for range predicates.
    - Observe literal value with "domain" of actual values
    - Can spot stale columns statistics…
    - Eg. HIGH2KEY = 2006-11-01, you've got 2007 data in the table…
    - WHERE SALES_DT BETWEEN 2007-01-01 AND 2007-01-31

67

Annotation predicate close-up

```
( SYSADM.LINEITEM.L_ORDERKEY = SYSADM.ORDER.O_ORDERKEY
                                                        COLCARDF=1,063,455/1,500,00
                                                        MAX_FREQ=0.0001645%/0.000066666666
                                                        FF=6.666666649834951E-7
  AND SYSADM.SUPPLIER.S_SUPPKEY = SYSADM.LINEITEM.L_SUPPKEY
                                                        COLCARDF=10,000/10,000
                                                        MAX_FREQ=99.99%/0.0119%
                                                        FF=9.99999901978299E-5
  AND SYSADM.LINEITEM.L_RETURNFLAG = 'R'
                                                        COLCARDF=4
                                                        MAX_FREQ=50.75460606218732%
                                                        FF=0.24650996923446655
  AND SYSADM.ORDER.O_ORDERDATE < ( DATE( '1993-10-01' ) + 3 MONTHS )
                                                        COLCARDF=2,304
                                                        MAX_FREQ=0.0414%
                                                        FF=0.30288922786712646
                                                        LKEY=1992-01-02  HKEY=1998-08-01
  AND SYSADM.ORDER.O_ORDERDATE >= DATE( '1993-10-01' )
                                                        COLCARDF=2,304
                                                        MAX_FREQ=0.0414%
                                                        FF=0.7351873517036438
                                                        LOW2KEY=1992-01-02
                                                        HIGH2KEY=1998-08-01
  AND SYSADM.SUPPLIER.S_NATIONKEY IN ( 1, 2, 3, 4, 5 )
                                                        COLCARDF=25
                                                        MAX_FREQ=4.21%
                                                        FF=0.1526399850845337
  AND SYSADM.SUPPLIER.S_SUPPKEY < 10000
                                                        COLCARDF=10,000
                                                        MAX_FREQ=99.99%
                                                        FF=0.9998999834060669
                                                        LOW2KEY=2
                                                        HIGH2KEY=9999
```


General annotation comments

- SQL analysis
  - Start at the table level
    - Identify any tables where qualified rows estimate looks off
  - Examine predicates on that table
    - Any individual predicates look too
      - Optimistic?
      - Pessimistic?
      - Frequencies, histograms?
    - After individual predicates are OK, any unaddressed correlation?
      - KEYCARD, COLGROUP?
  - Difficult to estimate predicate?
    - WHERE SUBSTR(C1,1,4) = ‘ABCD’

Query annotation transformed SQL

- Transformed SQL
  - All original SQL annotation benefits plus
  - Show optimizer rewritten SQL
    - Predicate transitive closure
    - Predicate transformations
    - Distribution and pruning
    - Merge / materialization

Query annotation transformed SQL

- Transformed SQL
  - Show SQL before / after distribution
  - Before / after pruning

Untransformed UIV SQL

```
SELECT X.DISCOUNT
      , SUM( X.QUANTITY ) AS SUMQ
      , AVG( X.QUANTITY * X.PRICE ) AS AVGP
  FROM SYSADM.LITEM_UIV_NOGBY AS X
 WHERE ( X.QUANTITY <= 5
       AND X.DISCOUNT > 0.08
       AND X.ORDERKEY > 30000000
     )
 GROUP BY X.DISCOUNT
```

- Where predicates referencing the view

72

Untransformed UIV - union all legs

```
FROM SYSADM.LITEM_UIV_NOGBY AS X
   ( SELECT SYSADM.LINEITEM.L_ORDERKEY
1  UNION ALL
     SELECT SYSADM.LINEITEM.L_ORDERKEY
2  UNION ALL
     SELECT SYSADM.LINEITEM.L_ORDERKEY
3  UNION ALL
     SELECT SYSADM.LINEITEM.L_ORDERKEY
4  UNION ALL
     SELECT SYSADM.LINEITEM.L_ORDERKEY
5  UNION ALL
     SELECT SYSADM.LINEITEM.L_ORDERKEY
   )
WHERE ( X.QUANTITY <= 5
      AND X.DISCOUNT > 0.08
      AND X.ORDERKEY > 30000000
```

73

Transformed UIV SQL - union all legs

```
FROM TABLE
      ( SELECT SYSADM.LINEITEM.L_DISCOUNT
      1 UNION ALL
        SELECT SYSADM.LINEITEM.L_DISCOUNT
      2 UNION ALL
        SELECT SYSADM.LINEITEM.L_DISCOUNT
      3 UNION ALL
        SELECT SYSADM.LINEITEM.L_DISCOUNT
      )
   AS DSNWFQB(02)
GROUP BY DSNWFQB(02).DISCOUNT
```

74

Query report

- Easy to read report which shows
  - Database / table space
  - Table
  - Index
  - Indexed columns
  - Column group cardinality
- Quick reference
  - Statistics on objects
  - Statistics on indexed columns
  - KEYCARD statistics mapped into index

75

New SQL

```
SELECT <cols>
FROM SYSIBM.SYSPACKSTMT AS A
                , SYSIBM.SYSPACKAGE AS C
WHERE ( A.QUERYNO > -1
      AND C.COLLID = 'TEST'
      AND C.NAME = 'PGM1'
      AND A.COLLID = C.COLLID
      AND A.LOCATION = C.LOCATION
      AND A.NAME = C.NAME
      AND A.VERSION = C.VERSION      )
ORDER BY <cols>
```

76

Query report

```
TABLE_SPACE       NACTIVEF     PARTS        SEGSIZE      PG_SIZE
DSNDB06.SYSPKAGE  -1.0         1            4            4

     TABLE                CARDF        NPAGESF      TABNO   QUALROWS
     SYSIBM.SYSPACKSTMT   -1           -1           1       6.3061666

     INDEX            CLU   UR NLEAF        NLEVEL   CR      KEYCOLNAME   COLCARDF     MCARDF
     SYSIBM.DSNKSX01  Y     U  -1           -1       0.0     LOCATION     -1           -1
                                                             COLLID       -1           -1
                                                             NAME         -1           -1
                                                             CONTOKEN     -1           -1
                                                             SEQNO        -1           -1

  COLUMN_GROUPMCARDF
-------------------------------------------------------------------------------------------
```

- Looks like someone forgot to run RUNSTATS…
- Hey… this is MY database!

77

Statistics advisor says…

```
RUNSTATS TABLESPACE DSNDB06.SYSPKAGE
  TABLE(SYSIBM.SYSPACKSTMT)
  COLUMN(NAME,QUERYNO,VERSION,COLLID)
  TABLE(SYSIBM.SYSPACKAGE)
  COLUMN(NAME,VERSION,COLLID)
  INDEX(SYSIBM.DSNKSX01 KEYCARD
        ,SYSIBM.DSNKKX02 KEYCARD
        ,SYSIBM.DSNKKX01 KEYCARD)
SHRLEVEL CHANGE REPORT YES
```

78

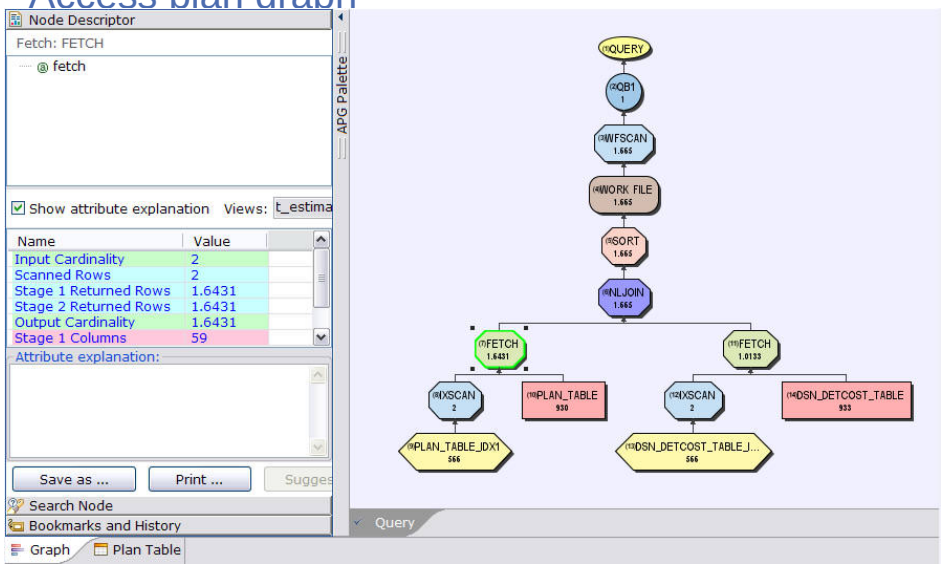
Query report after runstats…

```
TABLE_SPACE       NACTIVEF    PARTS       SEGSIZE     PG_SIZE
DSNDB06.SYSPKAGE  2160.0      1           4           4

      TABLE               CARDF       NPAGESF     TABNO  QUALROWS
      SYSIBM.SYSPACKSTMT  31040       1624        1      7.079398

      INDEX           CLU  UR  NLEAF       NLEVEL  CR     KEYCOLNAME  COLCARDF    MCARDF
      SYSIBM.DSNKSX01 Y    U   367         3       0.997  LOCATION    1           1
                                                          COLLID      43          43
                                                          NAME        118         149
                                                          CONTOKEN    -1          181
                                                          SEQNO       -1          31040

  COLUMN_GROUP                  MCARDF
(LOCATION,COLLID)               43.0
(LOCATION,COLLID,NAME)          149.0
(LOCATION,COLLID,NAME,CONTOKEN) 181.0
```

- Nice to see cascading MCARD's
  - Only matching on 2 or 3 columns?
  - Easy to see the KEYCARD bound…

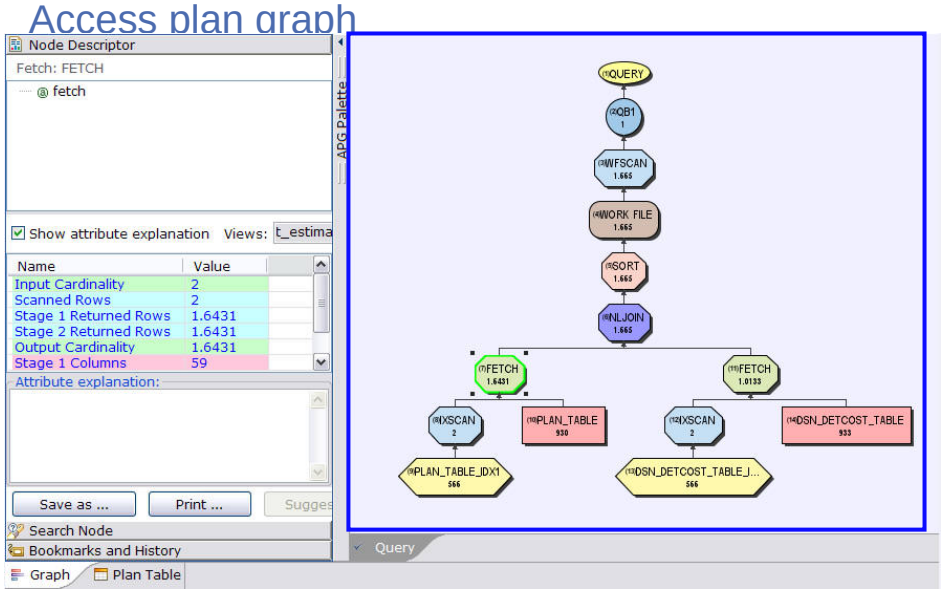
79

Access plan graph



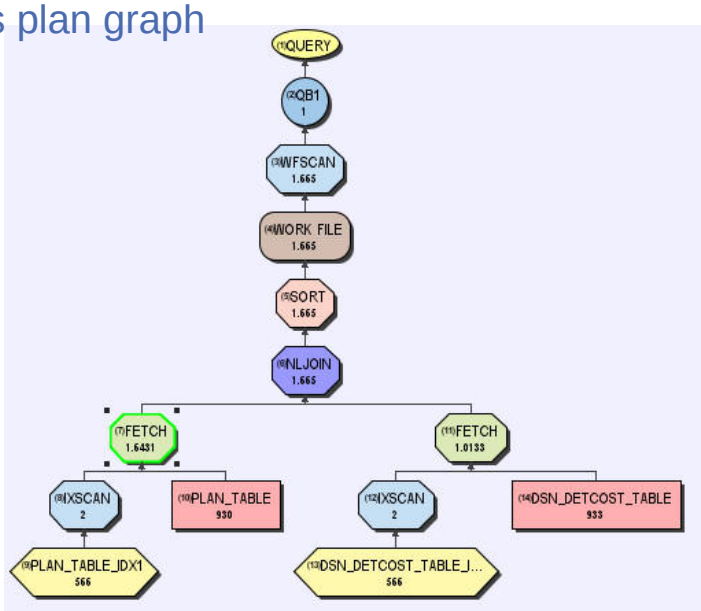
80

Access plan graph



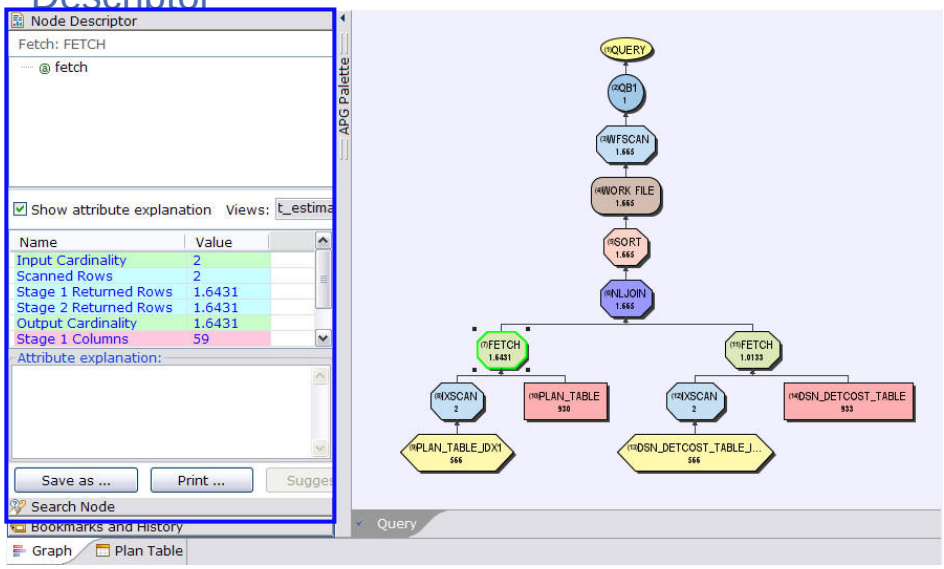
81

Access plan graph



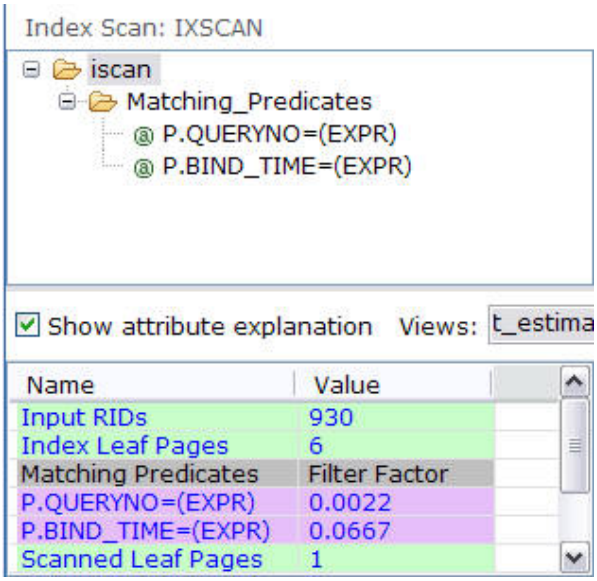
82

Descriptor



83

Descriptor



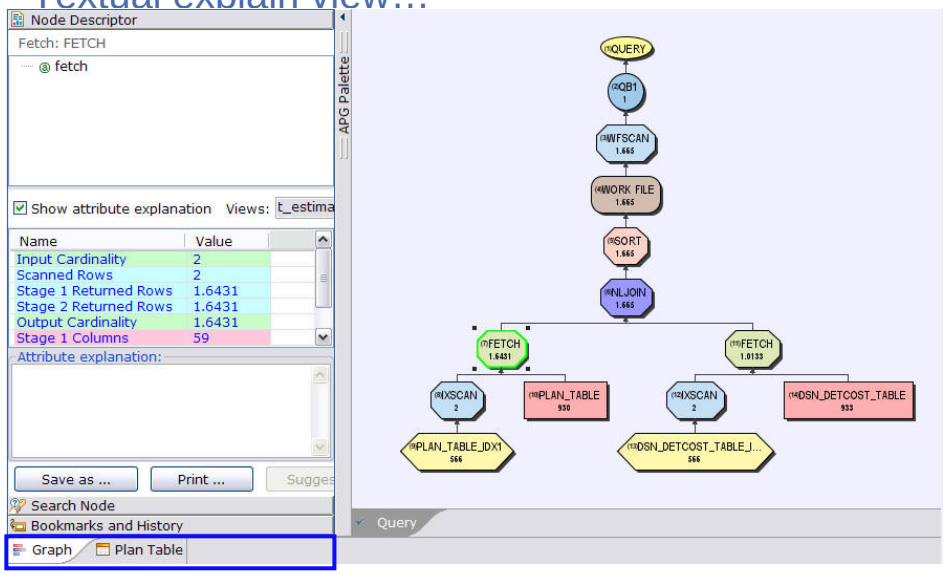
84

Loads of information…

- Information…
  - Identify matching, screening, S1, S2 predicates
  - Filter factors for predicates
  - Qualified partitions
  - Parallelism
    - Key range
    - Page Range
    - Degree of parallelism
  - Sort keys, sort key length
  - Estimated number of records at
    - Index matching, total index filtering
    - Data access
    - Sort input / output
    - Join

85

Textual explain view…



86

Textual explain view…

QUERYNO	QBLOCKNO	PLANNO	METHOD	CREATOR	TNAME
1099	1	1	0	OSCEXP	PLAN_TABLE
1099	1	2	1	OSCEXP	DSN_DETCOST_TABLE
1099	1	3	3		

- OSC shows plan_table rows
- Short / long format
- … this is obviously a truncated view…

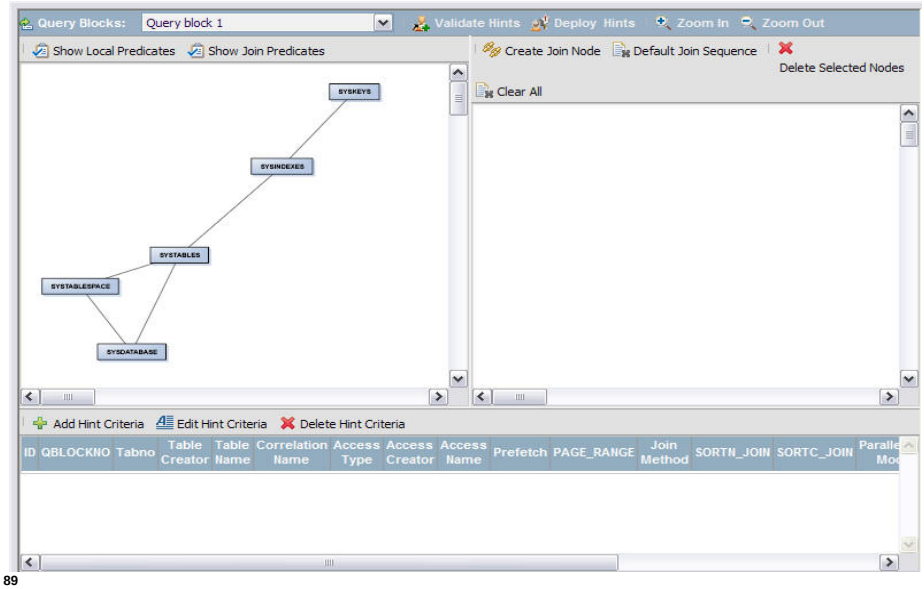
87

Visual Plan Hint

- Visual Plan Hint
  - Graphically alter the explain table
- Will show an overview
  - More details included as part of
  - Optimization hints presentation
    - http://www.bwdb2ug.org/PDF/Optimization%20Hints.pdf

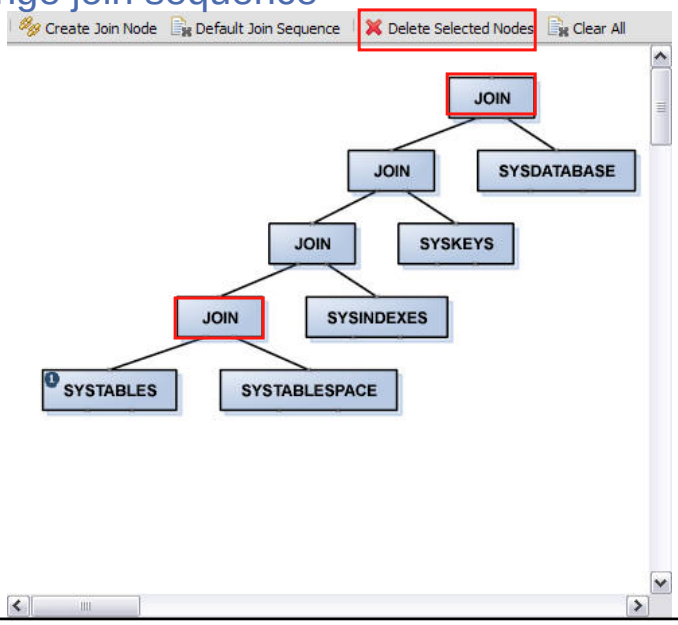
88

Visual Plan Hint



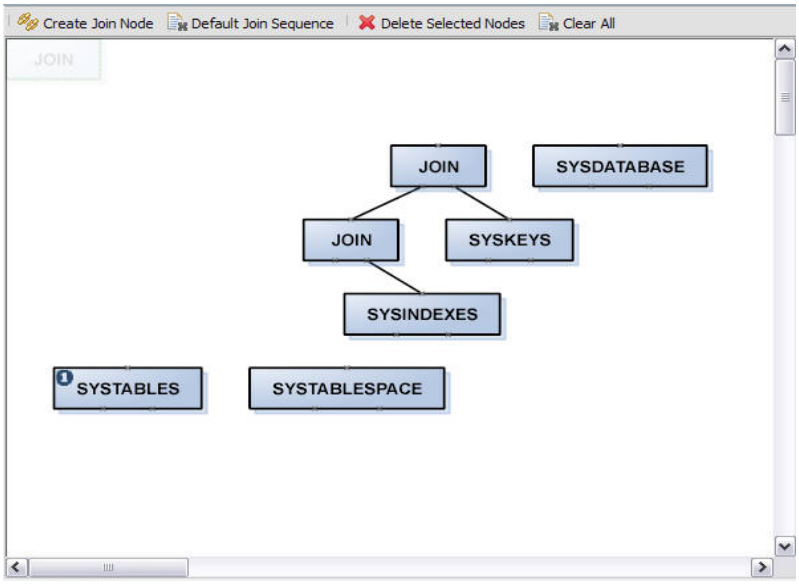
89

Change join sequence



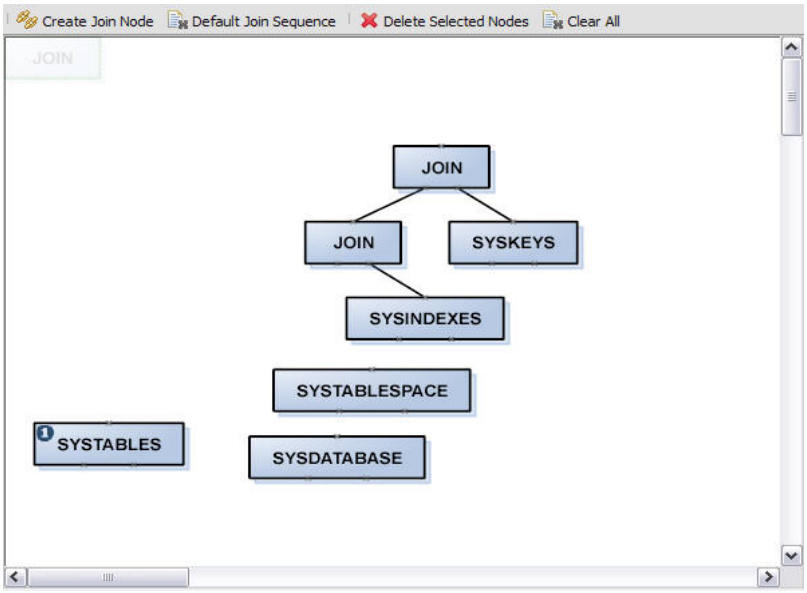
90

Join connectors removed



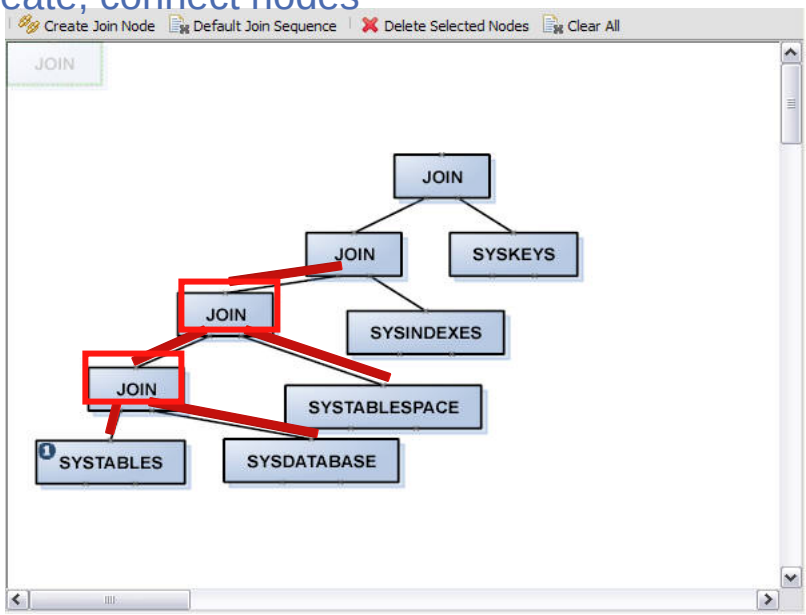
91

Move the tables around



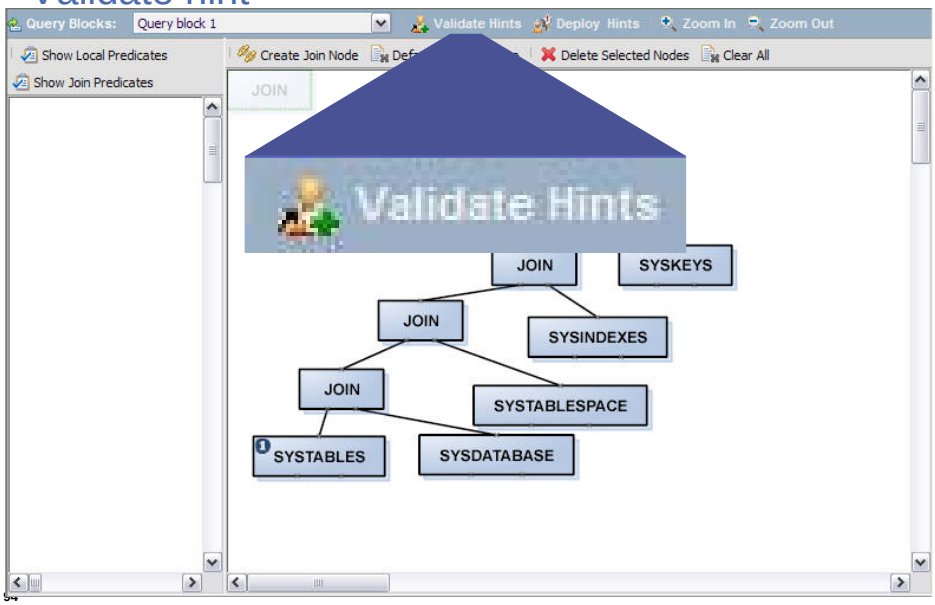
92

Create, connect nodes



93

Validate hint



Validate hint
(will show screen shot)

Plan table without using plan hint

QUERYNO	QBLOCKNO	APPLNAME	PROGNAME	PLANNO	METHOD	CREATOR	TNAME	TABNO	ACCESSTYPE	MATCHCOLS	ACCESSCREATOR	ACCESSNAME
1	1		SYSLH200	1	0	SYSIBM	SYSTABLES	1	I	2	SYSIBM	DSNDTX01
1	1		SYSLH200	2	1	SYSIBM	SYSTABLESPACE	2	I	2	SYSIBM	DSNDSX01
1	1		SYSLH200	3	1	SYSIBM	SYSINDEXES	3	I	2	SYSIBM	DSNDXX03
1	1		SYSLH200	4	1	SYSIBM	SYSKEYS	4	I	2	SYSIBM	DSNDKX01
1	1		SYSLH200	5	1	SYSIBM	SYSDATABASE	5	I	1	SYSIBM	DSNDDH01

Plan table after plan hint

QUERYNO	QBLOCKNO	APPLNAME	PROGNAME	PLANNO	METHOD	CREATOR	TNAME	TABNO	ACCESSTYPE	MATCHCOLS	ACCESSCREATOR	ACCESSNAME
15462	1		SYSLH200	1	0	SYSIBM	SYSTABLES	1	I	0	SYSIBM	DSNDTX02
15462	1		SYSLH200	2	1	SYSIBM	SYSDATABASE	5	I	1	SYSIBM	DSNDDH01
15462	1		SYSLH200	3	1	SYSIBM	SYSTABLESPACE	2	I	2	SYSIBM	DSNDSX01
15462	1		SYSLH200	4	1	SYSIBM	SYSINDEXES	3	I	2	SYSIBM	DSNDXX03
15462	1		SYSLH200	5	1	SYSIBM	SYSKEYS	4	I	2	SYSIBM	DSNDKX01

95

Visual Plan Hint Summary

- Visual Plan Hint
  - Graphically alter the explain table
  - Use of drop down lists reduces errors
  - Validation of optimization hint built in
  - Colorful compare of before / after access path, very nice

Conclusion

- Optimization Service Center / Optimization Expert
  - A valuable set of tools to define, analyze, and tune SQL workloads
  - Powerful advisors and diagnostic tools to assist with individual query diagnosis tuning.


Thank
You