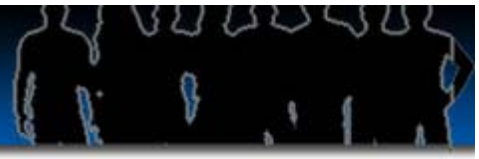
IDUG
The Worldwide DB2 User Community

IDUG DB2 Tech Conference
Denver, Colorado USA | May 2012

Advanced Query Tuning With IBM Data Studio for Developers

Tony Andrews
Themis Inc.



Objectives

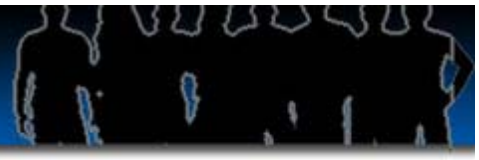
By the end of this presentation, you should:

- **Know how to use Data Studio to help improve query performance.**
- **Know the different access paths and understand how they are presented**
- **Understand filter factors**
- **Better understand how the DB2 optimizer determines access paths**
- **Better understand how to use and navigate Data Studio for SQL tuning**

Improving SQL Performance

- **System Tuning**
- **Change the SQL**
- **Gather / Alter Statistics**
- **Change Physical Design**

Themis inc.



Developers Should Focus On

- **Appropriate use of indexes**
- **Predicate Types**
- **Access Path Choice**
- **Filter Factors**
- **Known Statistics**
- **Clustering order**
- **Knowing ‘why’ any table space scan**
- **Stage 1 Predicates / Stage 2 / Residual**
- **Minimal Sorts**
- **Possible Rewrites**

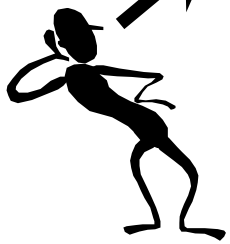
Themis inc.

Optimizer

Catalog Statistics

Object Definitions

Access Path



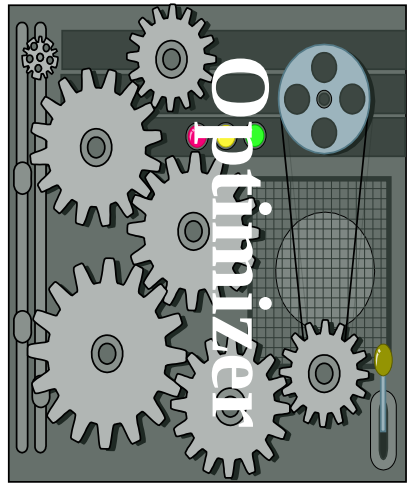
Access Path Hint

Rid Pools, Sort Pools
CPU Speed, # CPUs

Themis inc.

Explain

```
EXPLAIN PLAN SET QUERYNO = 10 FOR
SELECT LASTNAME,SALARY
    FROM EMP
WHERE EMPNO BETWEEN '000000' AND '099999'
    AND SALARY < 40000
```




```
OR
 BIND PACKAGE with option
        EXPLAIN(YES)
```

z/OS

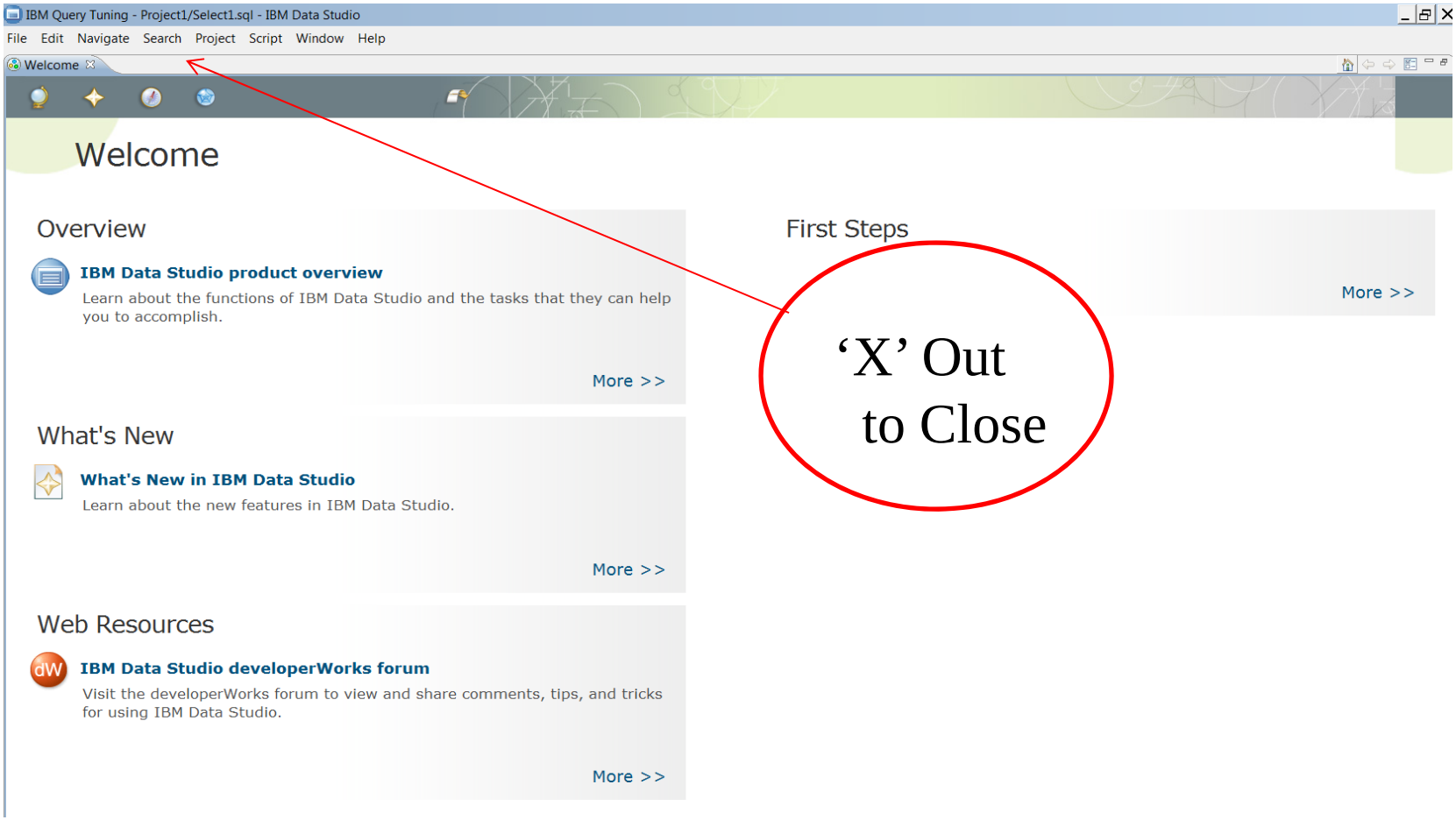
PLAN_TABLE
DSN_STATEMNT_TABLE
DSN_FUNCTION_TABLE
& a bunch of "other" tables

LUW

EXPLAIN_STATEMENT
EXPLAIN_PREDICATE
& a bunch of "other" tables

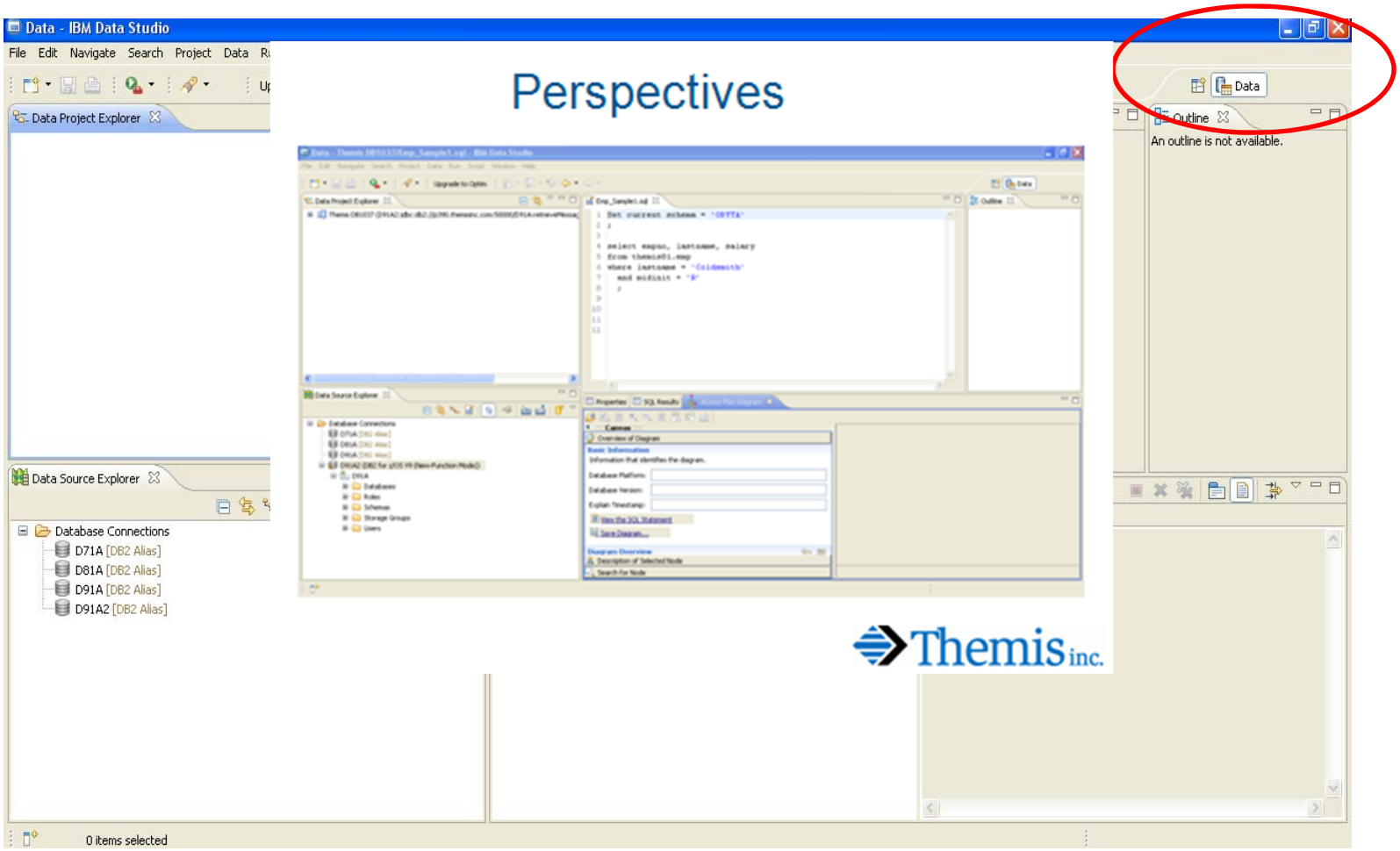
Themis inc.

IBM Data Studio



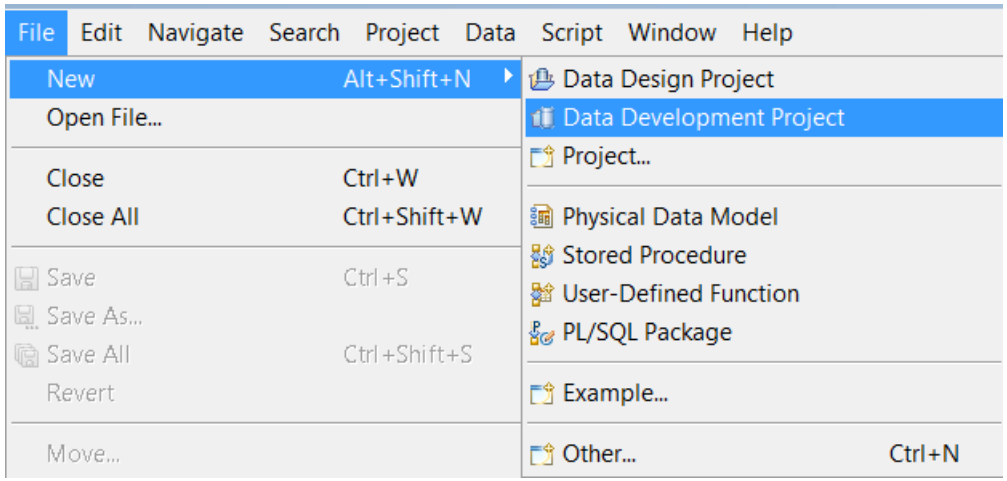
Themis inc.

IBM Data Studio



Themis inc.

Create a Project

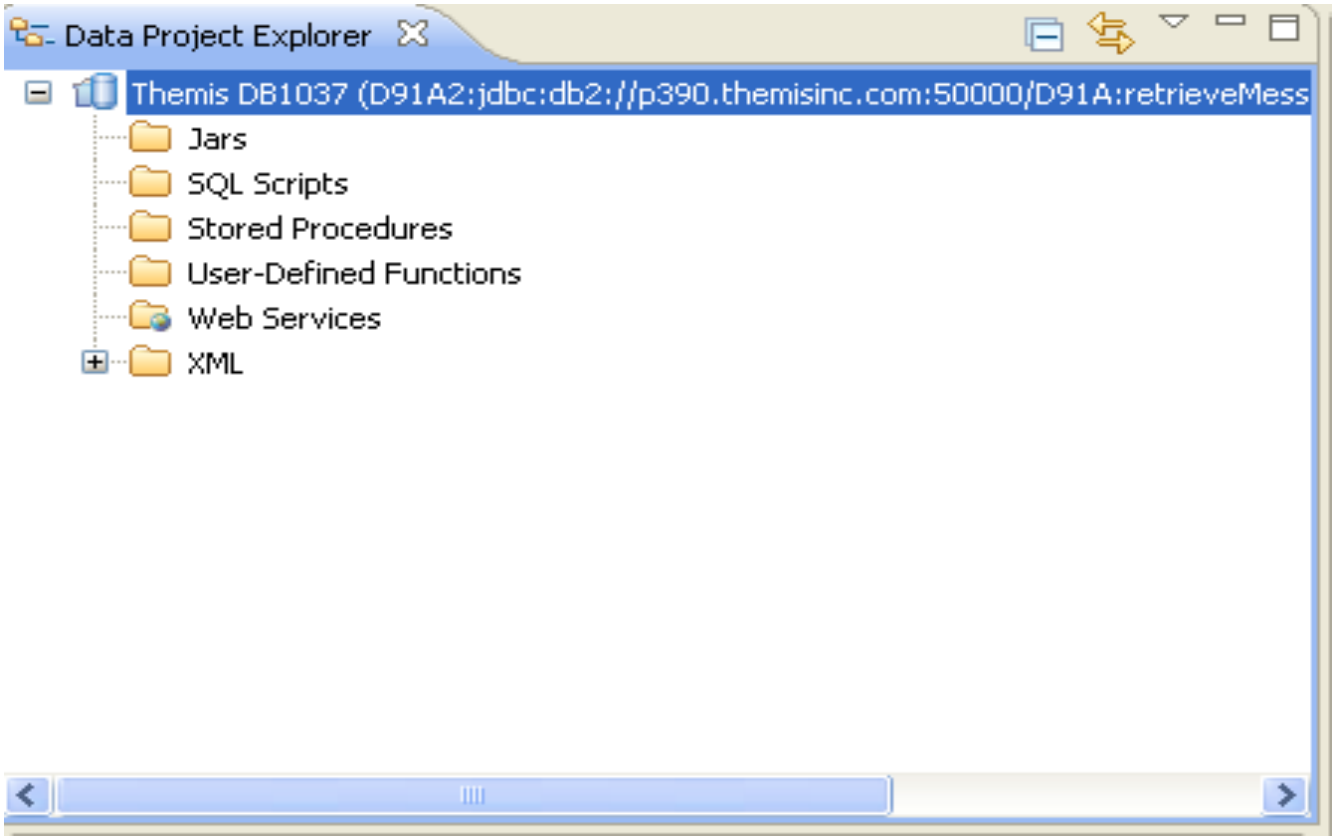


When Creating a Project

1) Assign a Project Name and Type (Data Devlp Project
1) Assign a Subsystem
2) Assign a Default Schema

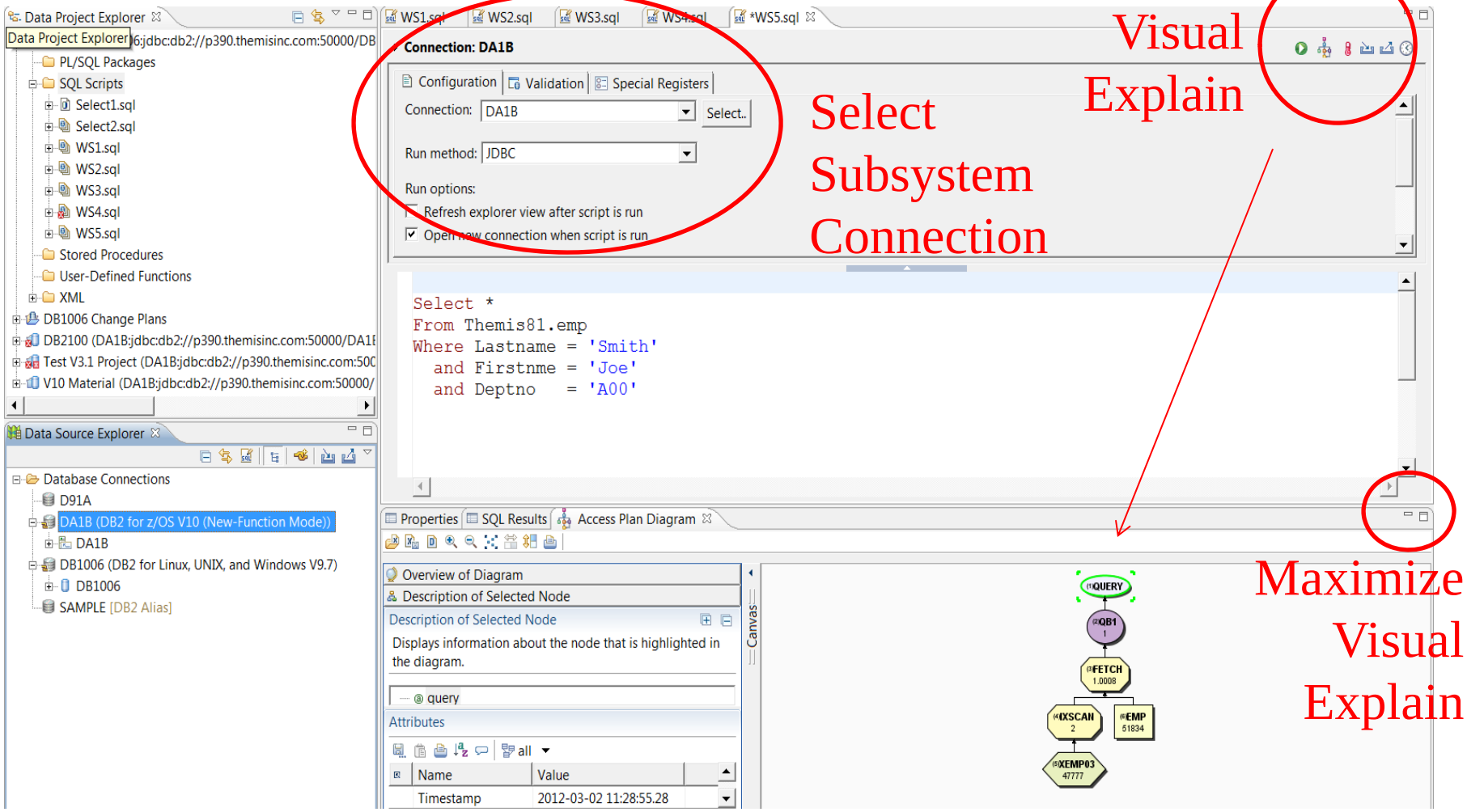
Themis inc.

Data Studio Project Created



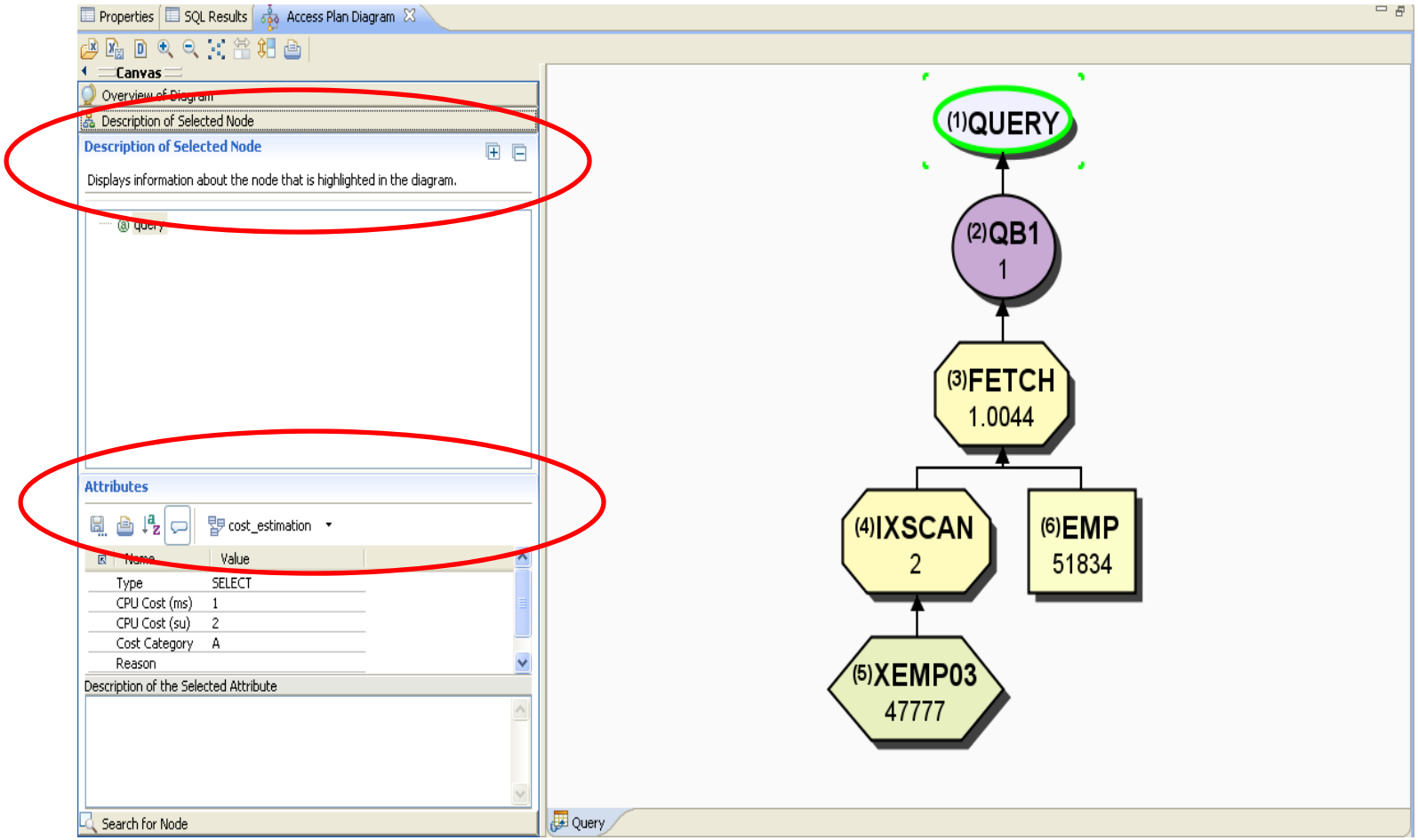
Themis inc.

Data Studio Explaining Queries



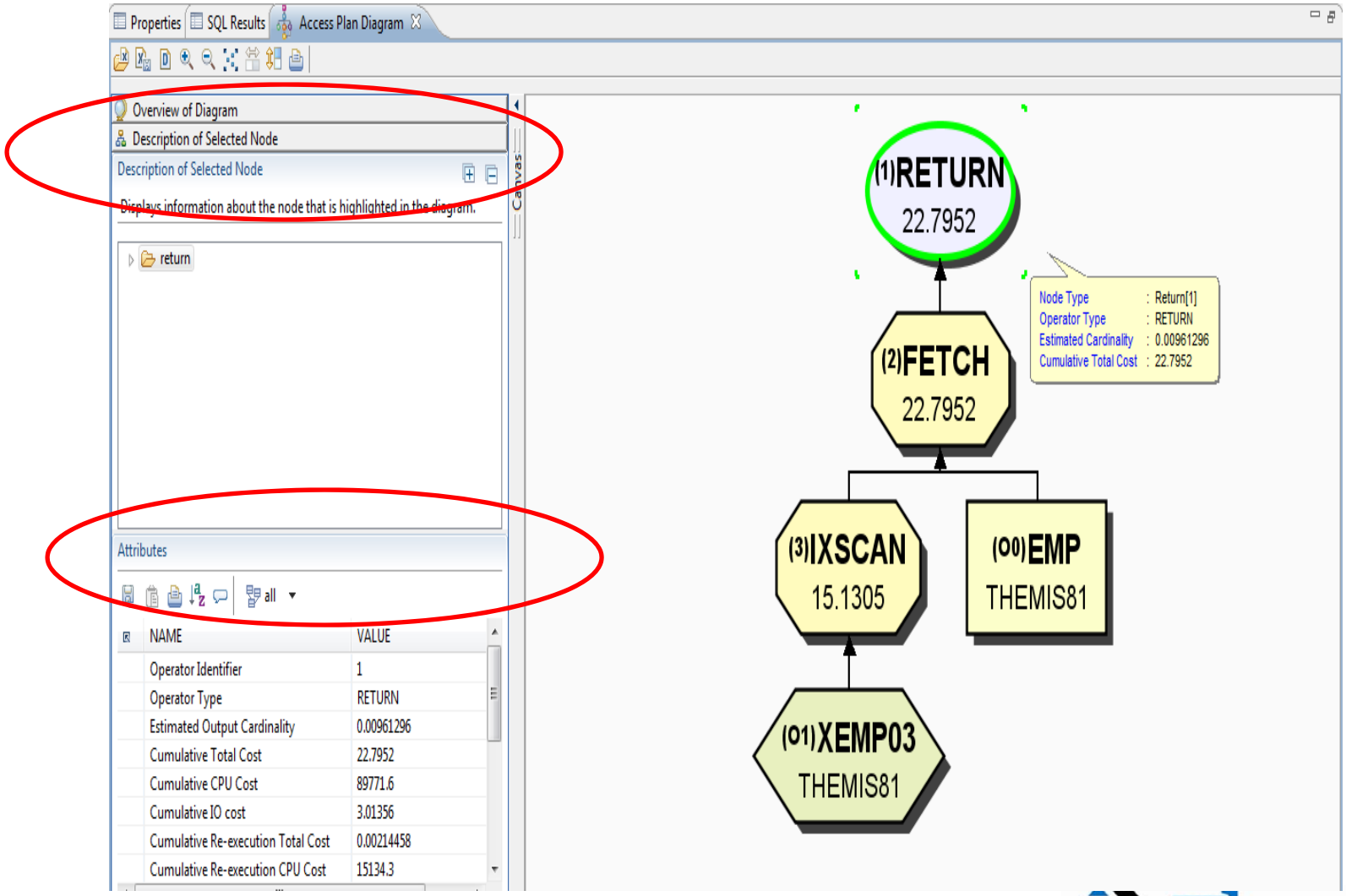
Themis inc.

z/OS Data Studio Access Path Graphs



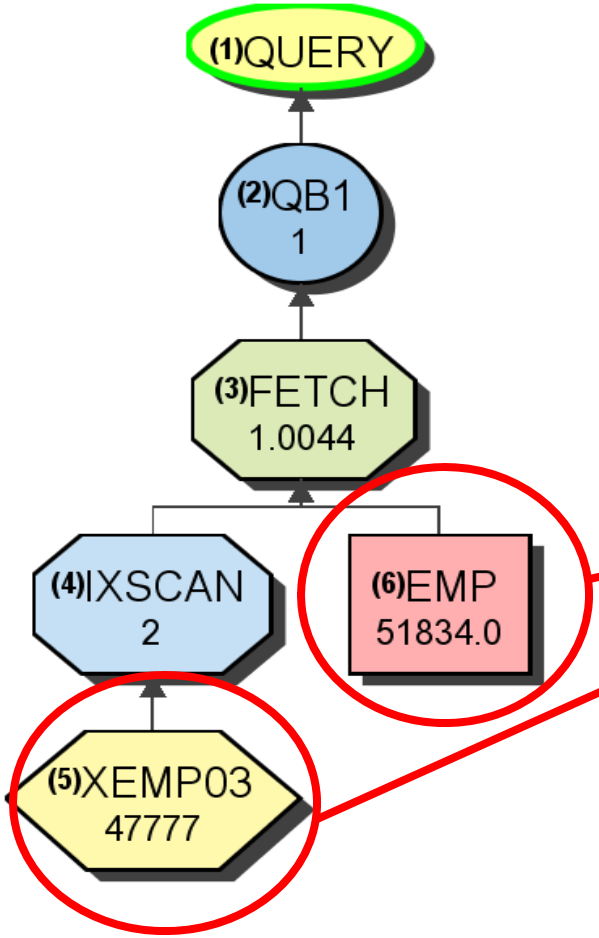
Themis inc.

LUW Data Studio Access Path Graphs



Themis inc.

Data Studio Access Path Graphs

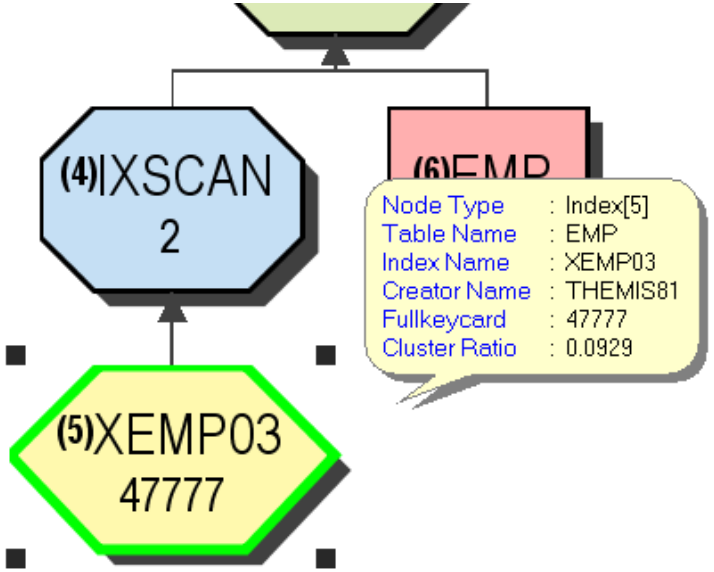


Sources of Data

Themis inc.

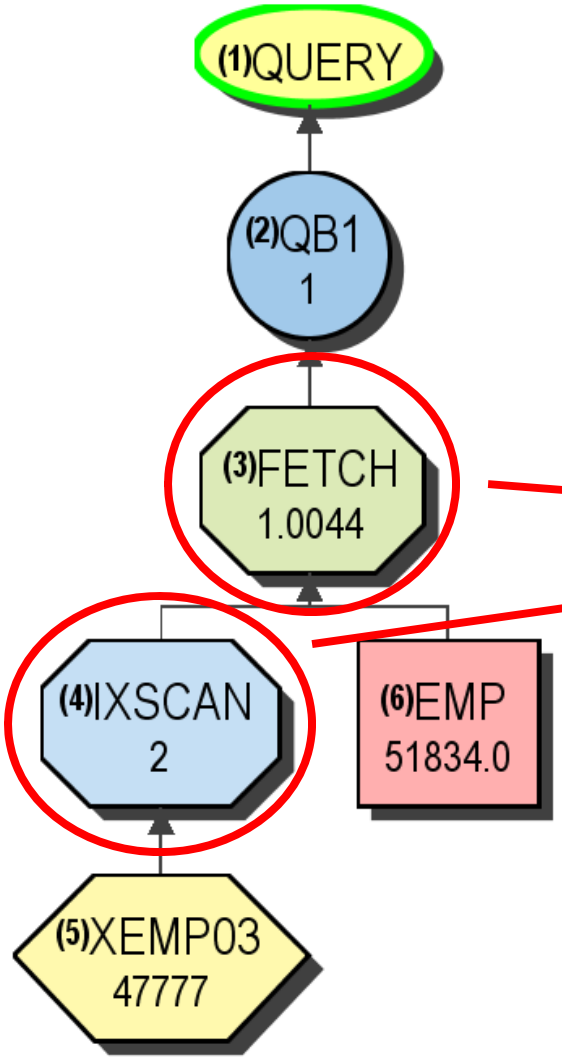
Data Studio Access Path Graphs

Name	Value
Table Name	EMP
Name	XEMP03
Creator	THEMIS81
Unique Rule	D
Clustering	N
Cluster Ratio	0.0929
First Key Cardinality	947
Full Key Cardinality	47777
Leaf Pages	549
Levels	3
Clustered	N
Type	2
Extension Type	
Padded	N
Compress Index	N
Data Repeat Factor	51609
Timestamp	2008-08-11 18:14:57.181709
Explain Time	2008-09-04 12:54:13.39



Themis inc.

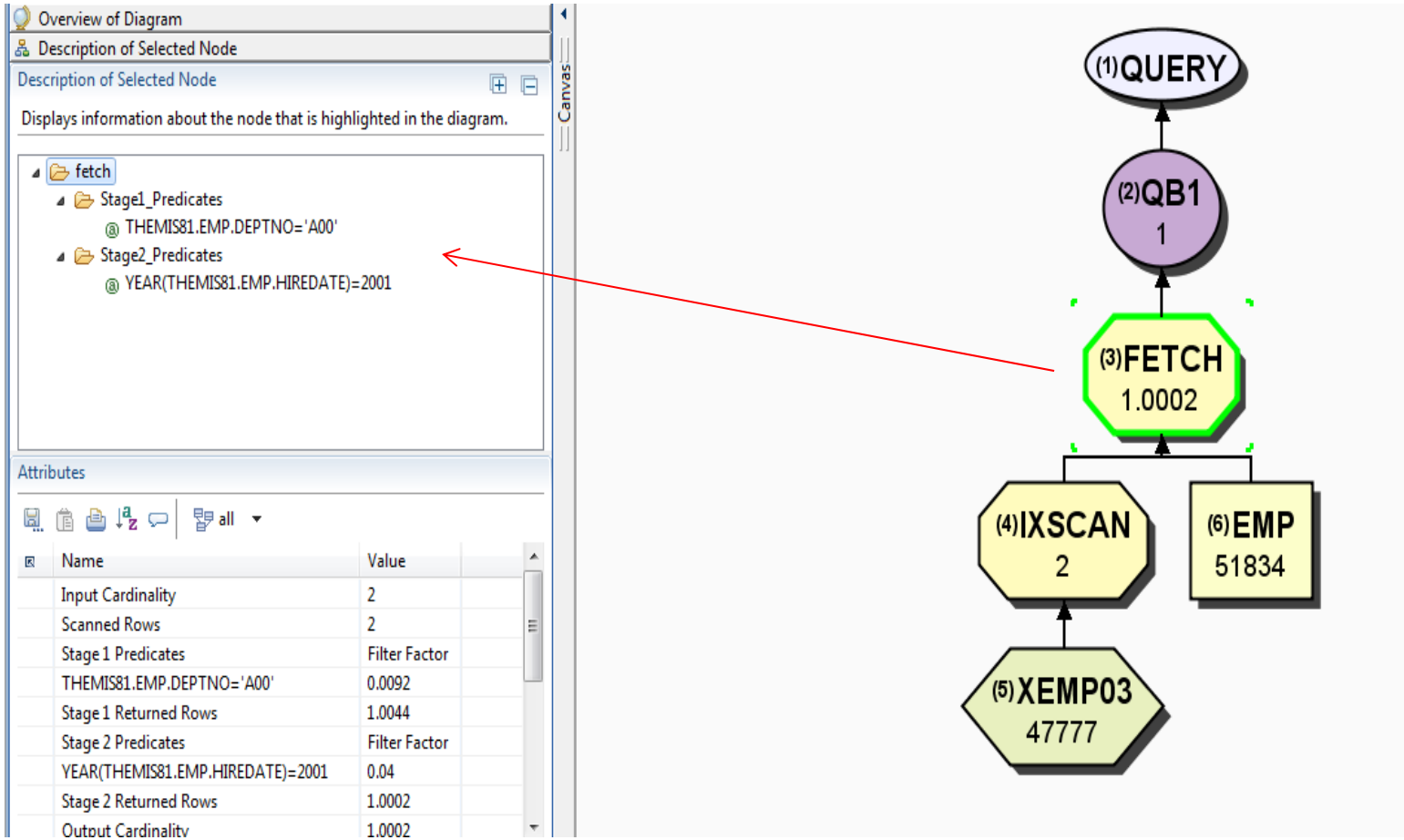
Data Studio Access Path Graphs



Data Retrieval Operations

Themis inc.

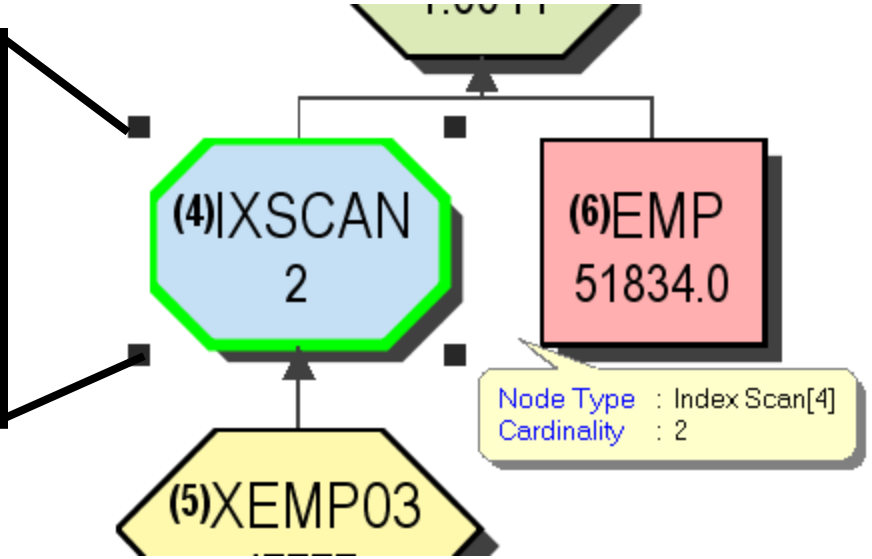
z/OS Stage 1 / 2 Predicates



Themis inc.

Data Studio Access Path Graphs

Input RIDs	51834
Index Leaf Pages	549
Matching Predicates	Filter Factor
THEMIS81.EMP.LASTNAME='Smith'	0.0024
THEMIS81.EMP.FIRSTNME='Joe'	0.0038
Scanned Leaf Pages	1
Output RIDs	2
Total Filter Factor	3.0149182E-5
Matching Columns	2



Themis inc.

Data Studio Access Path Graphs

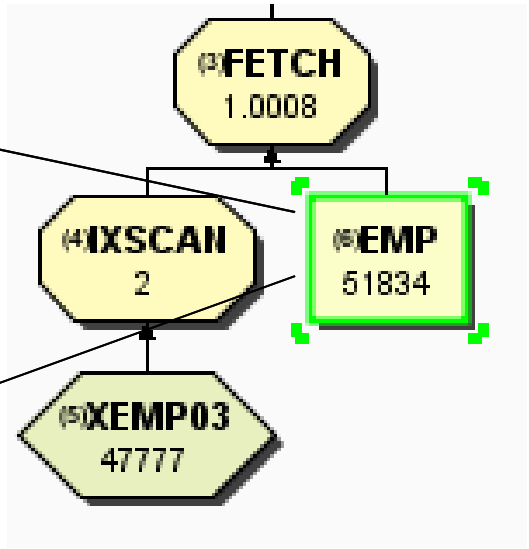
Displays information about the node that is highlighted in the diagram.

- table(EMP)
  - Columns
  - Indexes
  - Tablespace
  - Table_Partitions
  - ColGroup

Attributes

all

Name	Value
Name	EMP
Creator	THEMIS81
Correlation Name	
Type	T
Table Number	1
Qualifying Rows	1.0008
Base Table Type	T
Table Space	TS00EMP

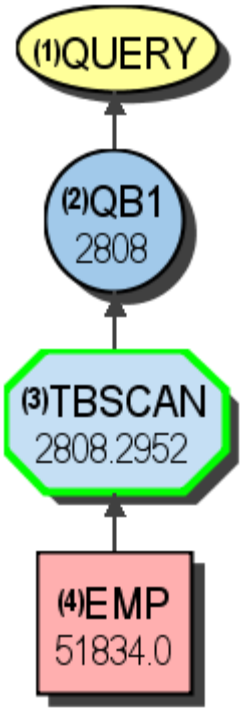


Themis inc.

Tablespace Scan

```
SELECT EMPNO, LASTNAME, SALARY
  FROM EMP
  WHERE EMPNO BETWEEN '000000' AND '099999'
   AND SALARY < 40000
```

PLAN NO	METHOD	TNAME	ACCESS TYPE	MATCH COLS	ACCESS NAME	INDEX ONLY	PREFETCH
1	0	EMP	R	0		N	S



Plan Table

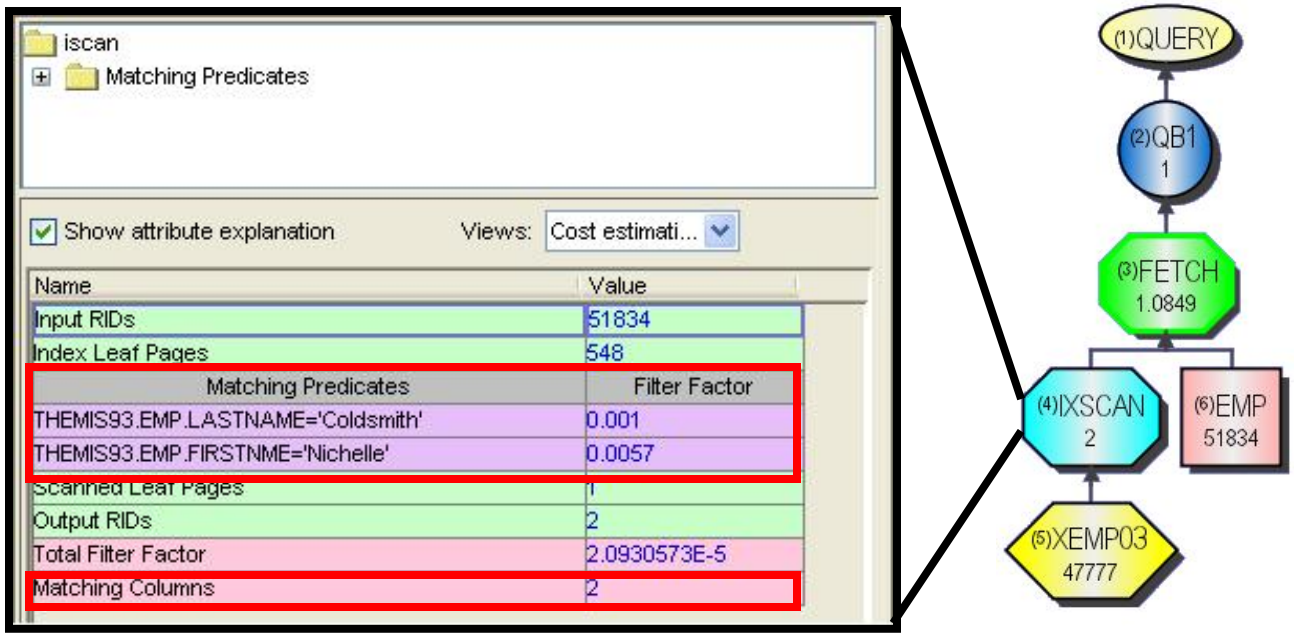
Data Studio Estimated number of rows

z/OS Index Scan - Matching

```
SELECT * FROM EMP
WHERE LASTNAME = 'Coldsmith'
AND FIRSTNME   = 'Nichelle';
```

PLAN_TABLE

PLAN NO	METHOD	TNAME	ACCESS TYPE	MATCH COLS	ACCESS NAME	INDEX ONLY	PREFETCH
1	0	EMP	I	2	XEMP03	N	



z/OS Index Screening

INDEX XEMP03 on
(LASTNAME, FIRSTNME, MIDINIT)

```
SELECT * FROM EMP
    WHERE LASTNAME = 'Coldsmith'
      AND MIDINIT = 'R';
```

Index Screening Predicate

PLAN_TABLE

PLAN NO	METHOD	TNAME	ACCESS TYPE	MATCH COLS	ACCESS NAME	INDEX ONLY	PREFETCH
1	0	EMP	I	1	XEMP03	N	

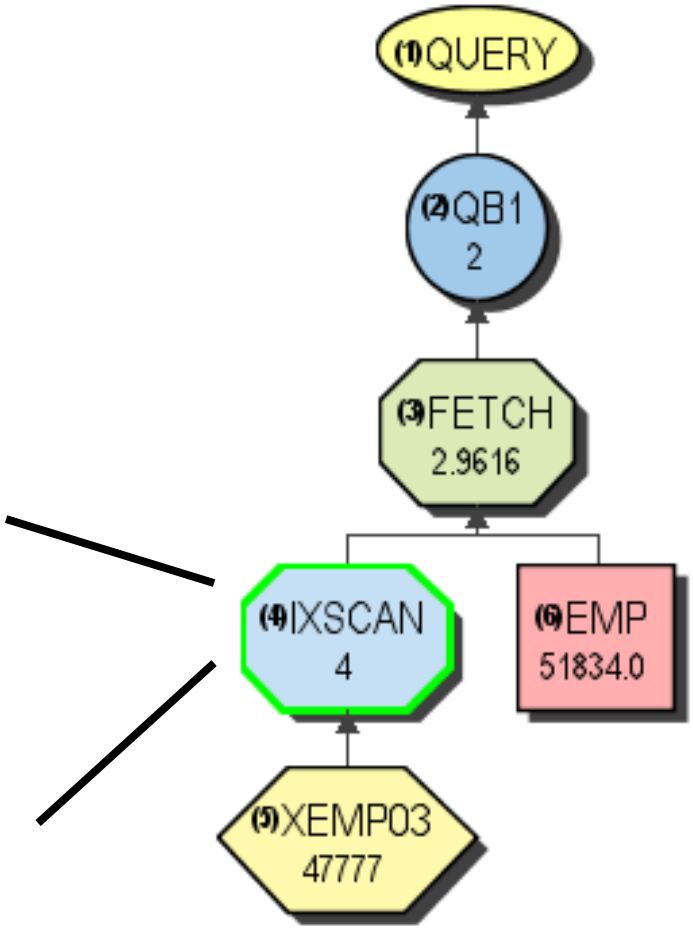
z/OS Index Screening (cont)

Index Scan: IXSCAN

- iscan
  - Matching_Predicates
    - @ THEMIS81.EMP.LASTNAME='Coldsmith'
  - Screening_Predicates
    - @ THEMIS81.EMP.MIDINIT='R'

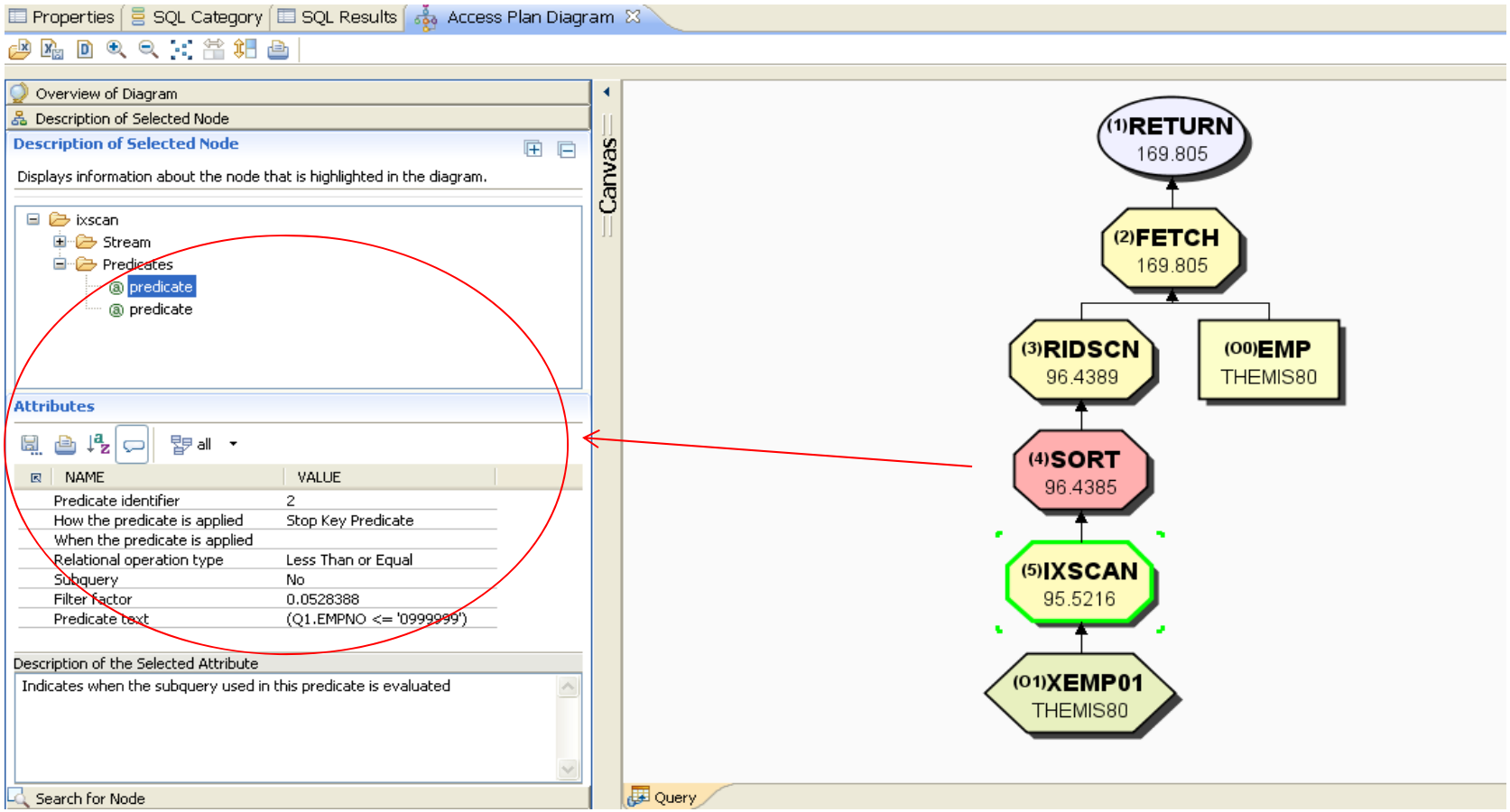
Show attribute explanation Views: cost_estimat

Name	Value
Input RIDs	51834
Index Leaf Pages	549
Matching Predicates	Filter Factor
THEMIS81.EMP.LASTNAME='Coldsmith'	0.001
Scanned Leaf Pages	1
Screening Predicates	Filter Factor
THEMIS81.EMP.MIDINIT='R'	0.037
Output RIDs	4
Total Filter Factor	5.8711856E-5
Matching Columns	1



LUW Index Scan - Matching (Start/Stop Keys)

```
SELECT * FROM EMP
WHERE EMPNO BETWEEN '000000' and '099999'
```



z/OS Index Scan - Nonmatching

```
SELECT * FROM EMP
WHERE FIRSTNME = 'Michelle'
   AND MIDINIT = 'R';
```

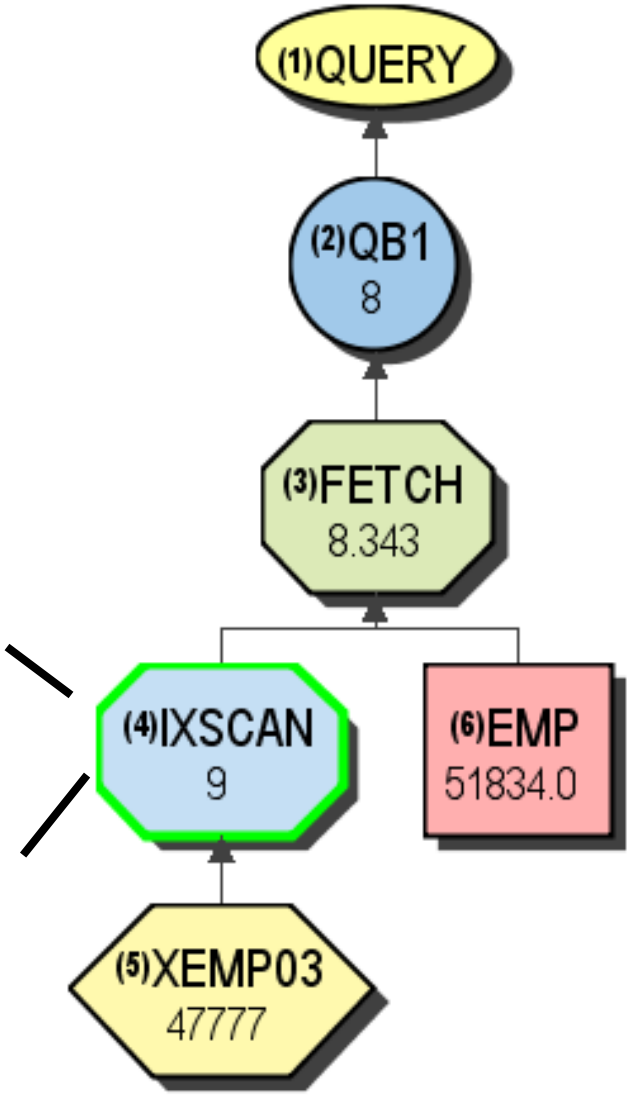
PLAN_TABLE

PLAN NO	METHOD	TNAME	ACCESS TYPE	MATCH COLS	ACCESS NAME	INDEX ONLY	PREFETCH
1	0	EMP	I	0	XEMP03	N	

z/OS Index Scan - Nonmatching

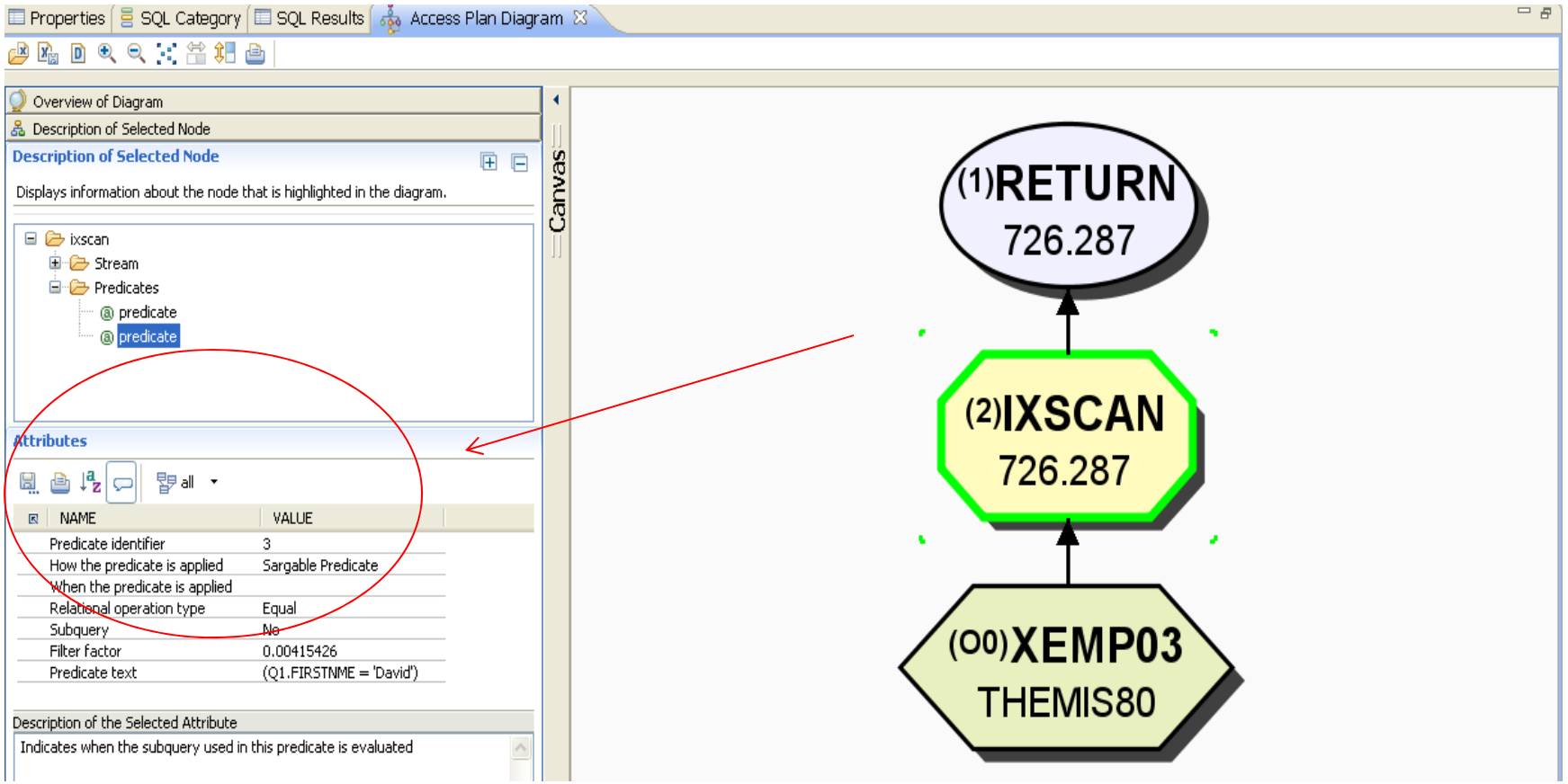
```
SELECT * FROM EMP
WHERE FIRSTNME = 'Michelle'
AND MIDINIT = 'R';
```

Name	Value
Input RIDs	51834
Index Leaf Pages	549
Scanned Leaf Pages	549
Screening Predicates	Filter Factor
THEMIS81.EMP.FIRSTNME='Nichelle'	0.0038
THEMIS81.EMP.MIDINIT='R'	0.037
Output RIDs	9
Total Filter Factor	0.0002
Matching Columns	0



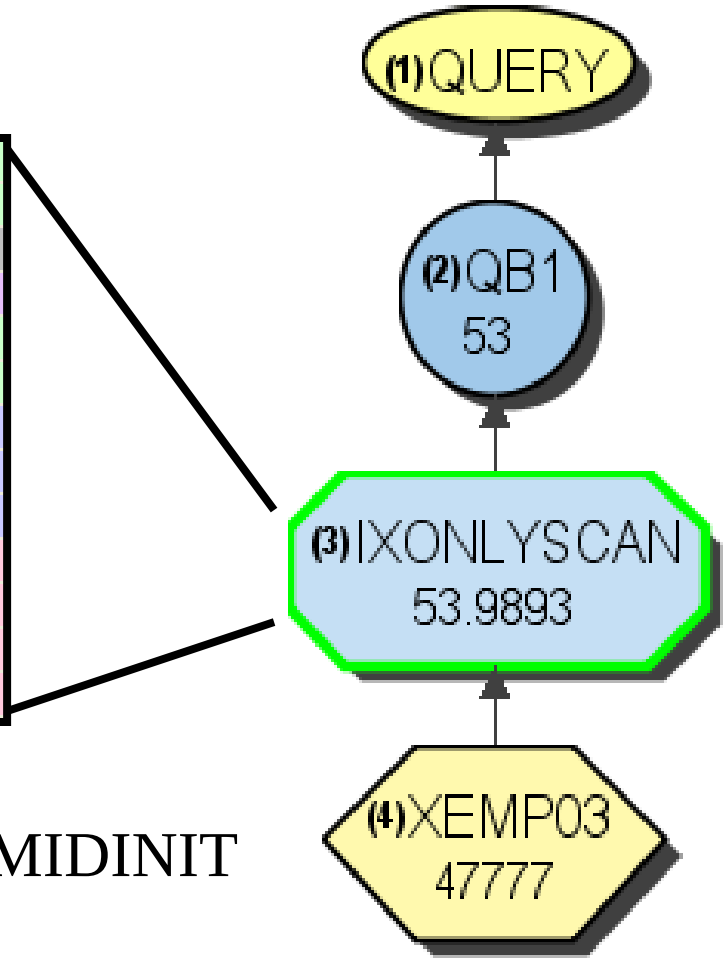
LUW Index Scan – Non Matching (Sargeable)

```
SELECT LASTNAME* FROM EMP
WHERE FIRSTNME = 'David' and MIDINIT = 'A'
```



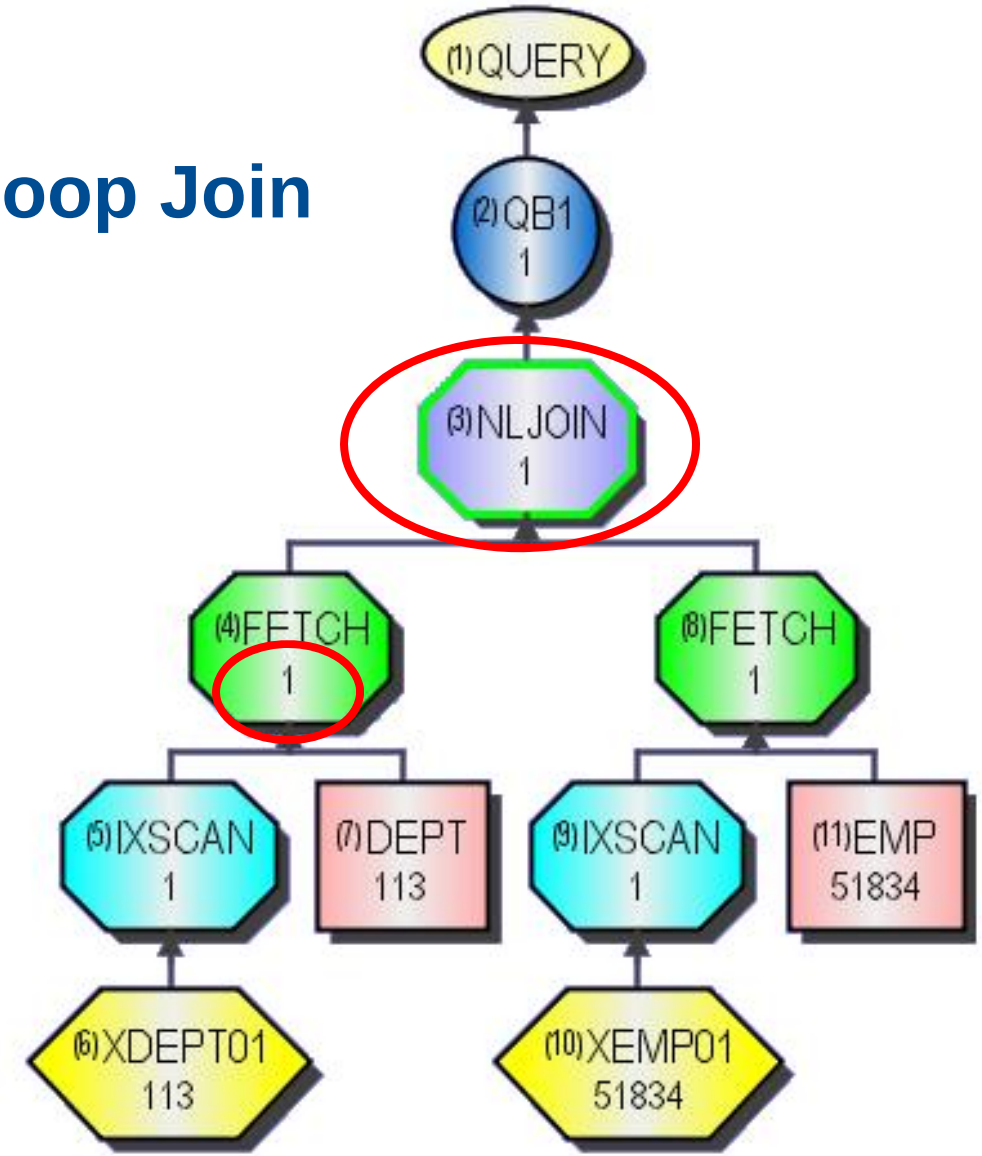
Index Only Access

Input RIDs	51834
Index Leaf Pages	549
Matching Predicates	Filter Factor
THEMIS81.EMP.LASTNAME LIKE 'Jo%'	0.001
Scanned Leaf Pages	1
Output RIDs	53.9893
Cumulative Total Cost	N/A
Cumulative IO Cost	N/A
Cumulative CPU Cost	N/A
Matching Filter Factor	0.001
Total Filter Factor	0.001
Prefetch	
Matching Columns	1

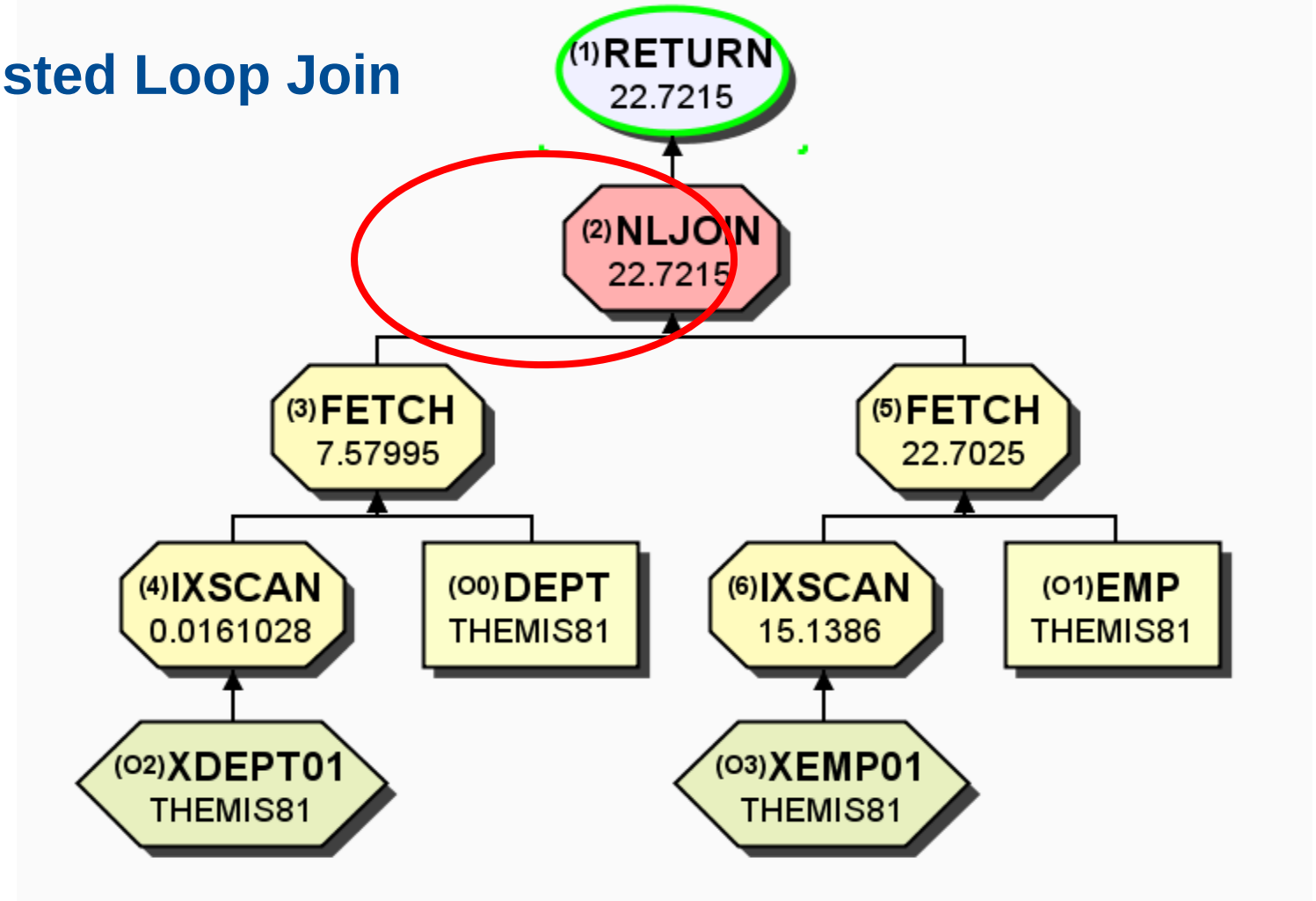



```
SELECT LASTNAME, FIRSTNME, MIDINIT
FROM EMP
WHERE LASTNAME LIKE 'Jo%'
```

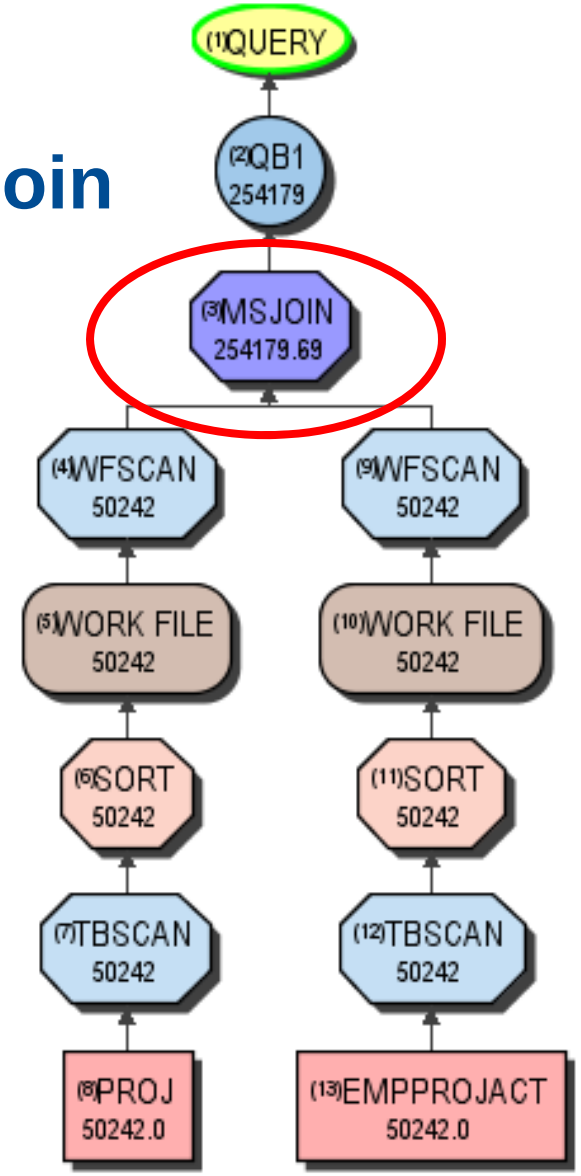
z/OS Nested Loop Join



LUW Nested Loop Join

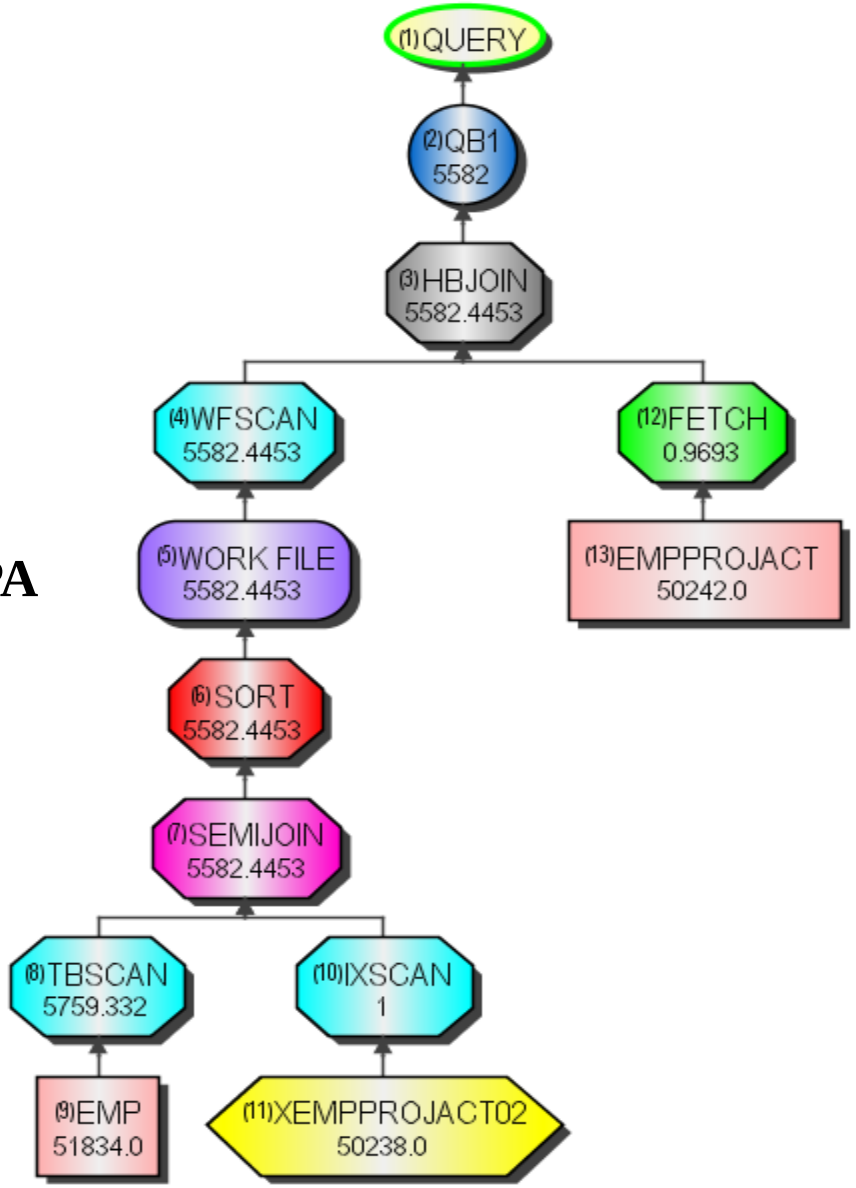


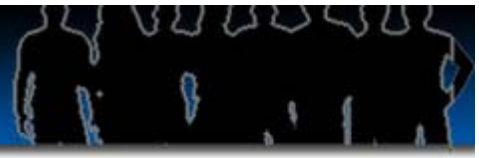
z/OS Merge Scan Join



z/OS Hybrid Join

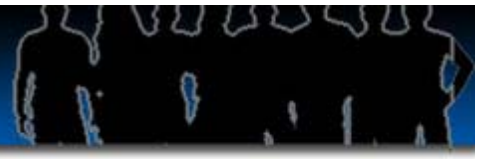
SELECT LASTNAME, PROJNO
FROM EMP E JOIN EMPPROJACT EPA
ON E.EMPNO = EPA.EMPNO
WHERE E.JOB = 'FIELDREP'





Which Join Method

1) Depends on the predicates
2) How much filtering on the tables
3) Possible indexes
4) Optimization level
5) Clustering of table data



Sort Activities

Data Sorts

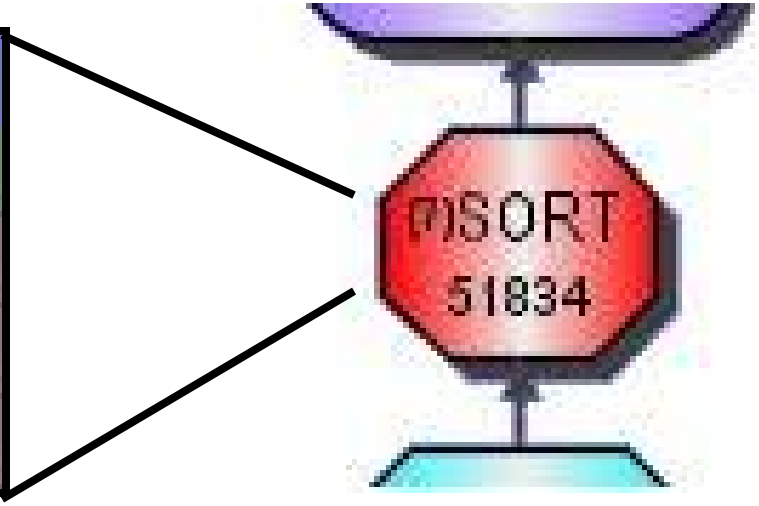
- ✓ORDER BY
- ✓GROUP BY
- ✓DISTINCT
- ✓UNION
- ✓Subqueries
- ✓JOIN

RID Sorts

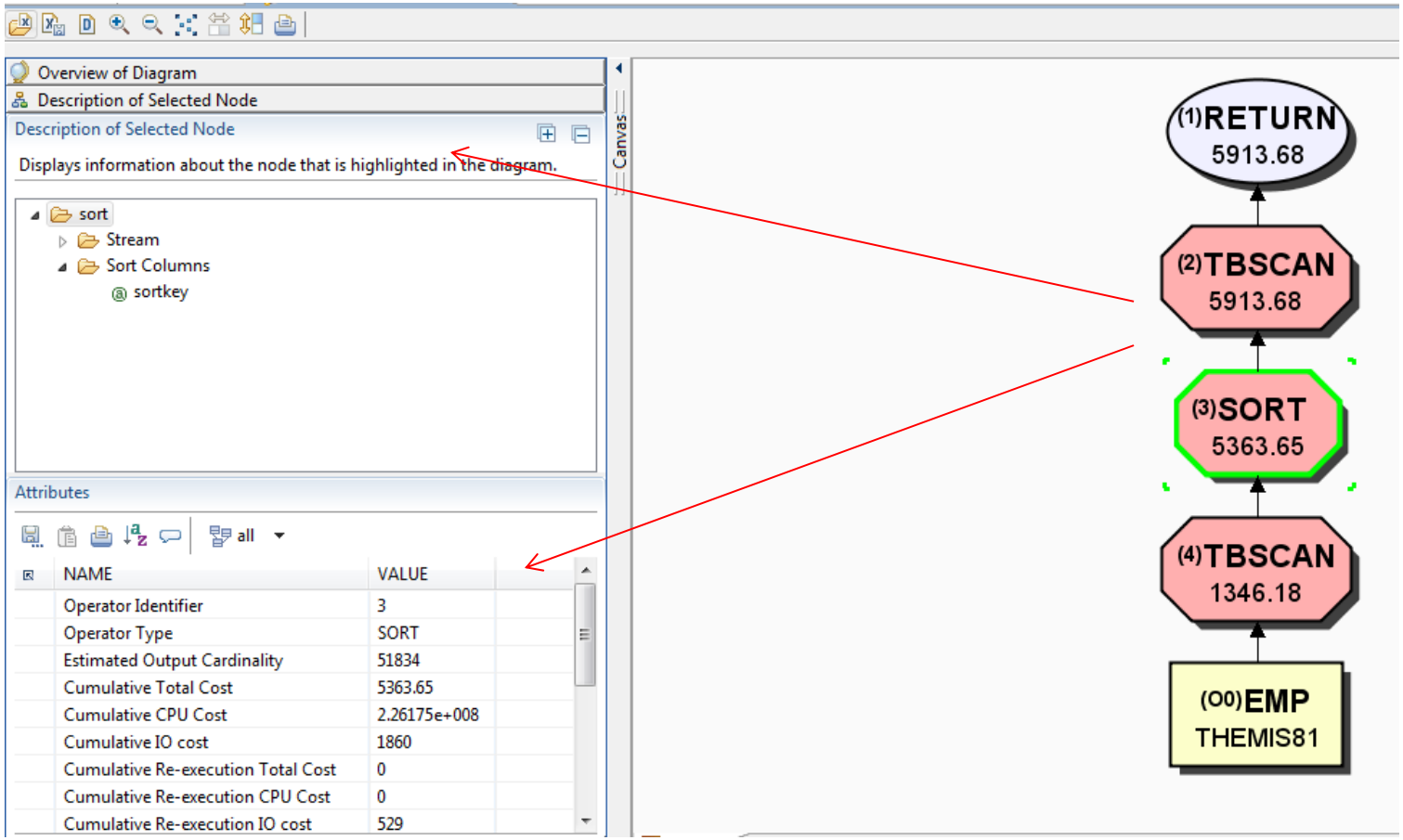
- ✓ List Prefetch
- ✓ Multiple Index Access
- ✓ Hybrid Join

z/OS Data Sorts via Data Studio

Input Cardinality	51834
Output Cardinality	51834
Pages	489
Record Size	21
Key Size	4



LUW Data Sorts via Data Studio



Predicate Generation Through Transitive Closure

The Premise

If A must equal B

And A must be RED,

Then B must also be RED.

Predicate Generation Through Transitive Closure Cont'd

Single Table DB2 Generated Predicate

Index XDEPT1 on DEPTNO
Index XDEPT3 on ADMRDEPT

```
SELECT . . . .
  FROM DEPT
  WHERE DEPTNO = ADMRDEPT
  AND ADMRDEPT = 'A00';
```

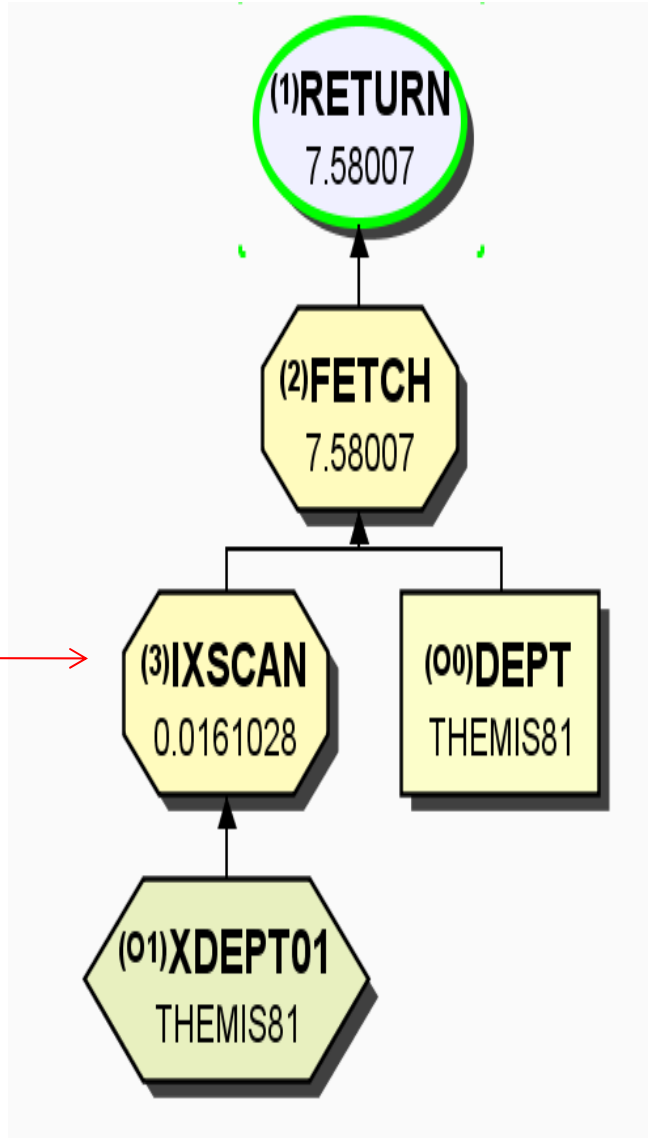
```
SELECT . . . .
 FROM DEPT
 WHERE DEPTNO = ADMRDEPT
 AND ADMRDEPT = 'A00'
 AND DEPTNO = 'A00' ;
```

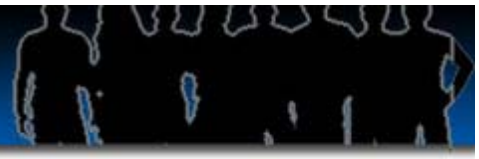
XDEPT1 index chosen !

Predicate Transitive Closure

```
SELECT . . . .
  FROM DEPT
  WHERE DEPTNO = ADMRDEPT
  AND ADMRDEPT = 'A00';
```

Note: Index on DEPTNO chosen





Predicate Transitive Closure

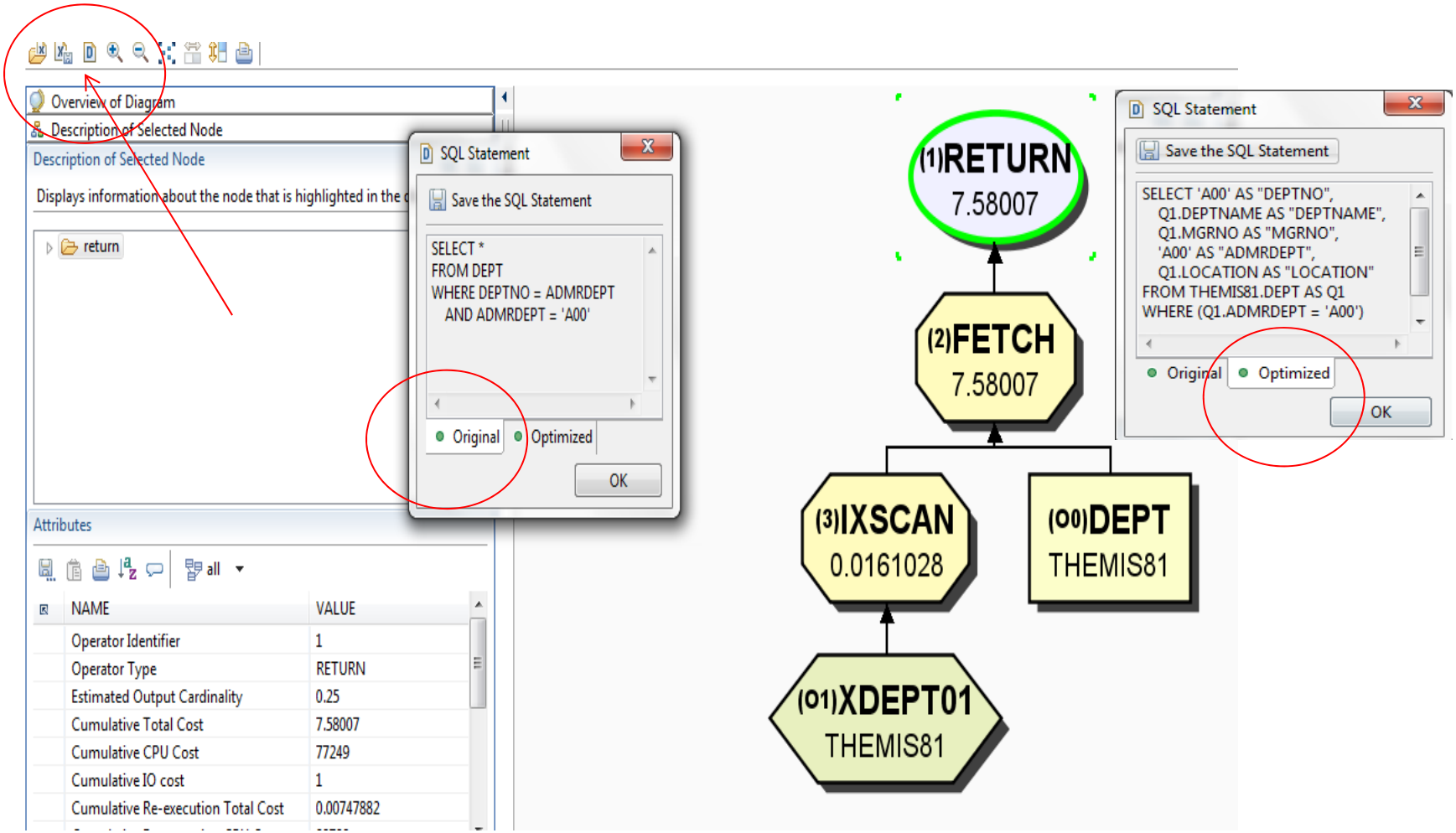
z/OS:

Transitive closure takes place for all predicates other than LIKE (as of V10)

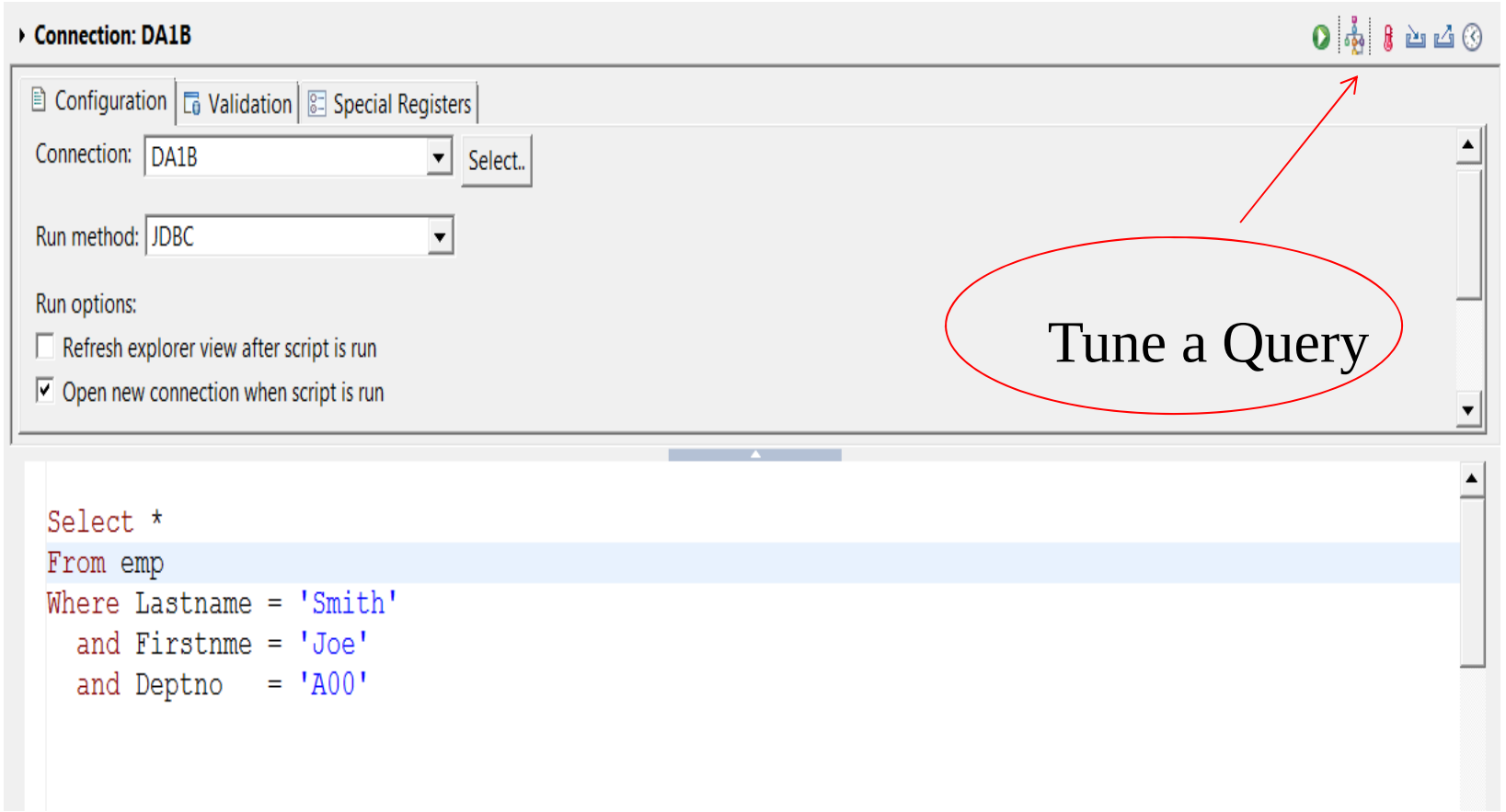
LUW

Transitive closure only takes place for EQUAL predicates. Developers should code for RANGE, LIKE, IN, BETWEEN, …)

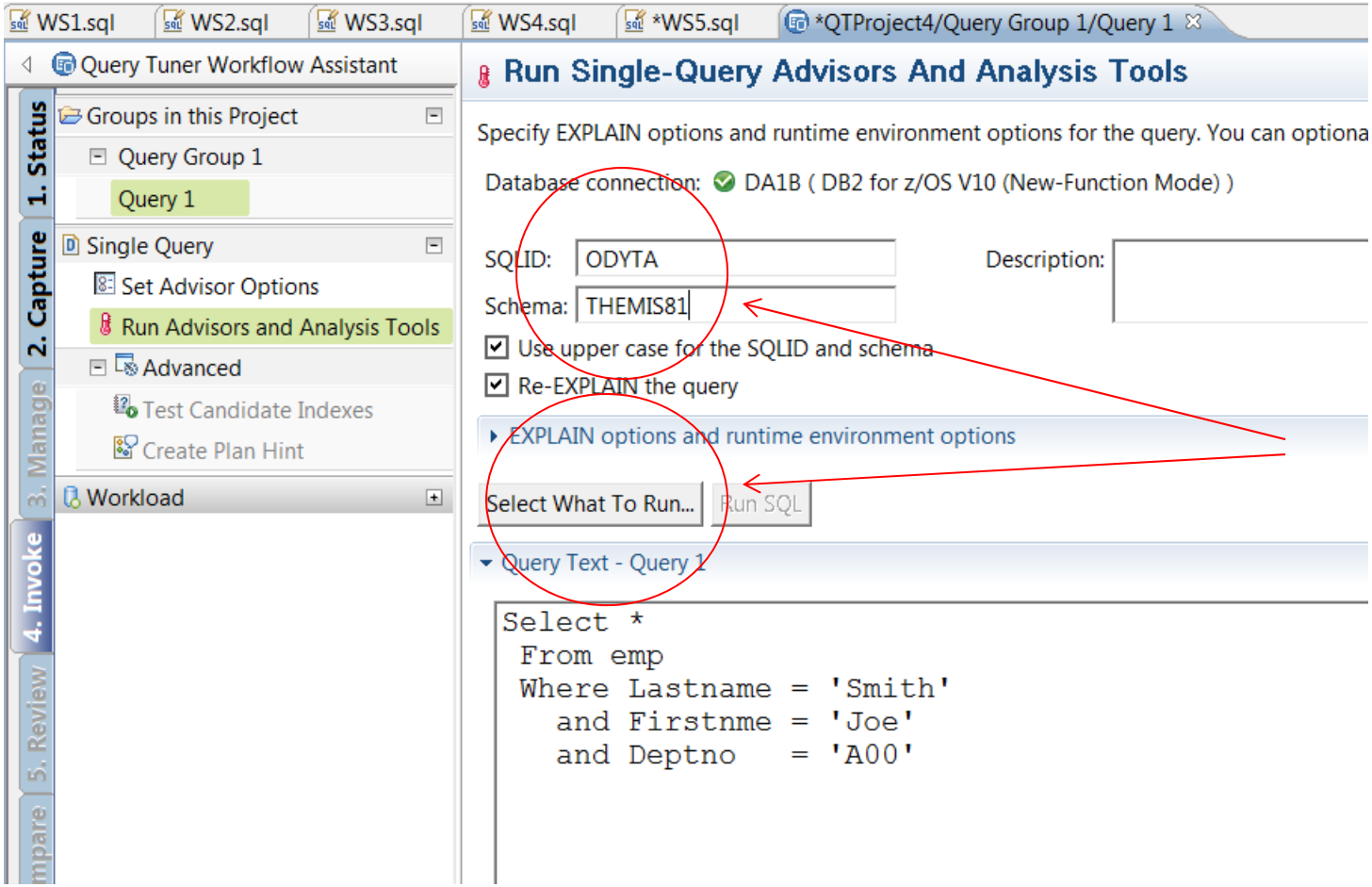
LUW Predicate Transitive Closure



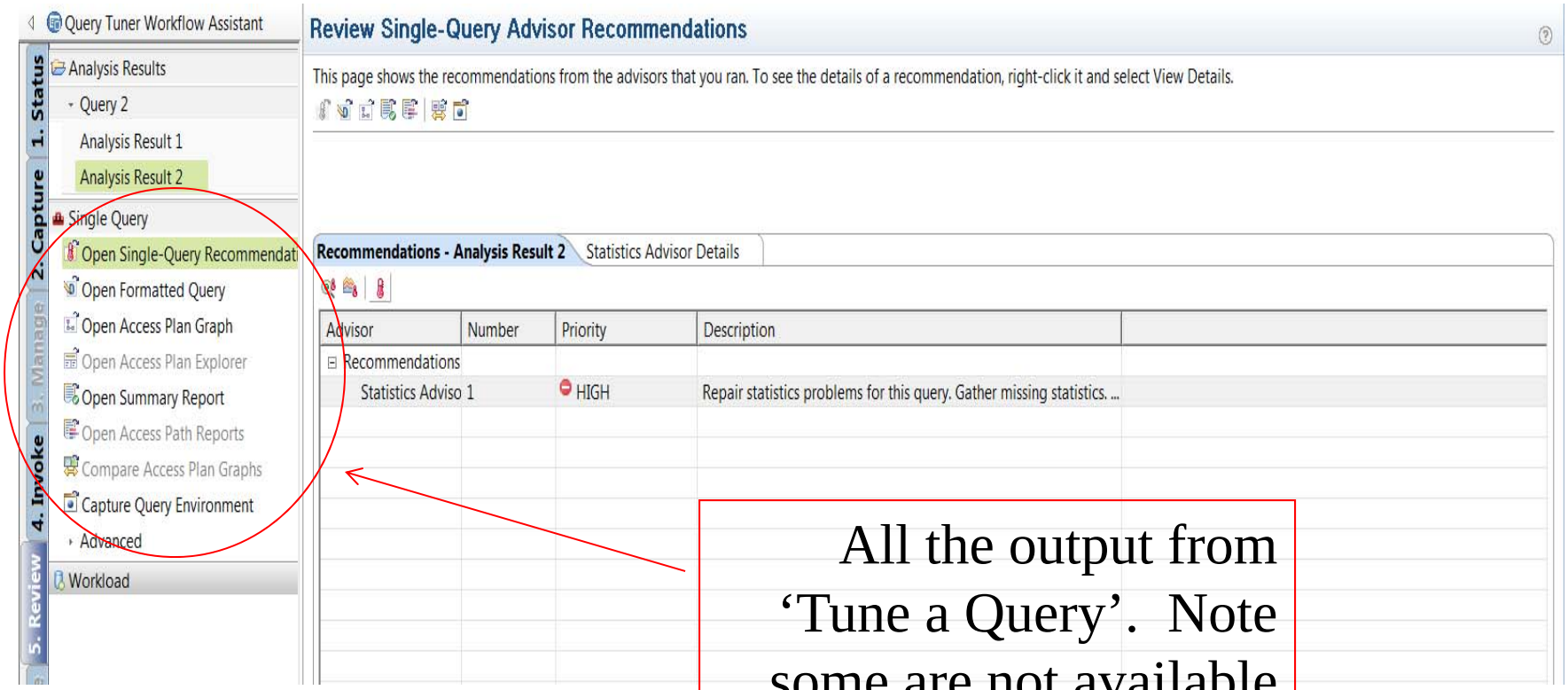
Tuning a Query



Tuning a Query

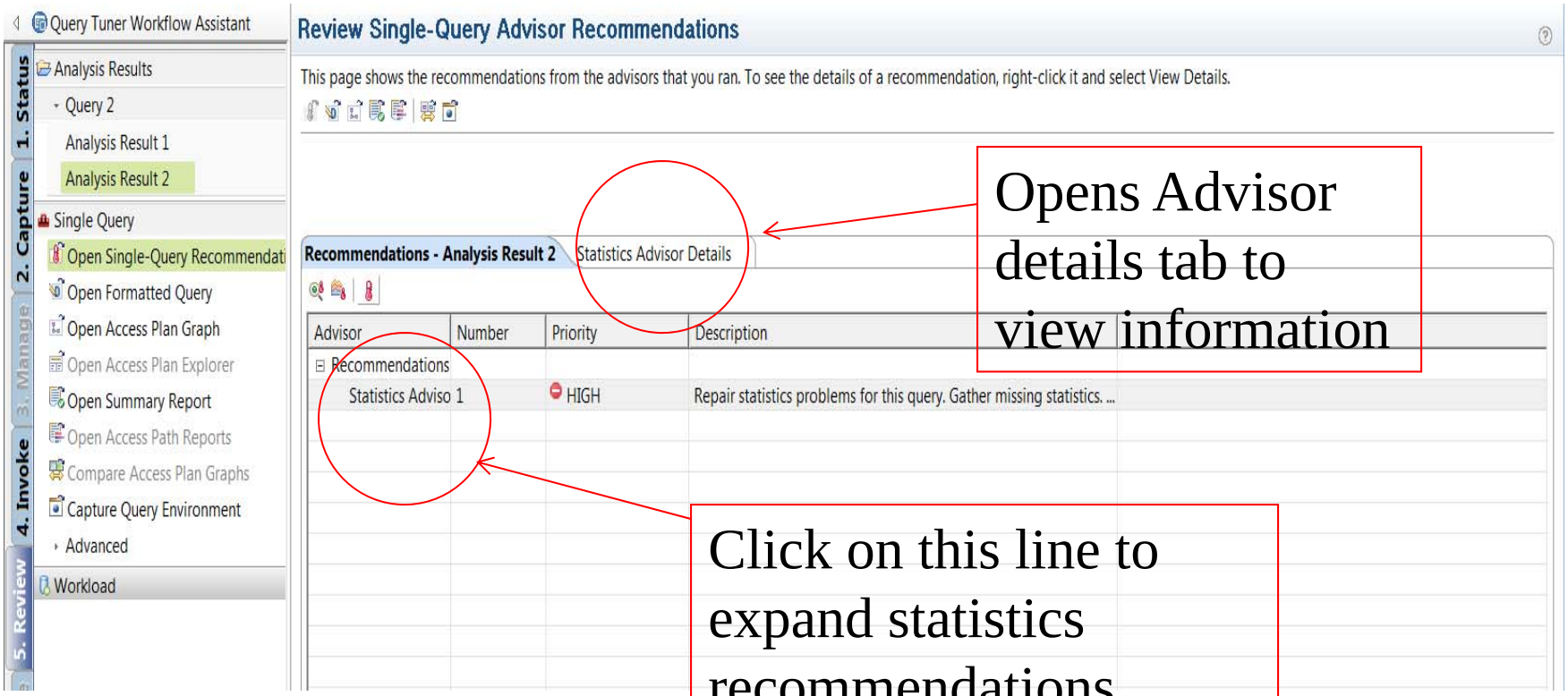


Tuning a Query Output

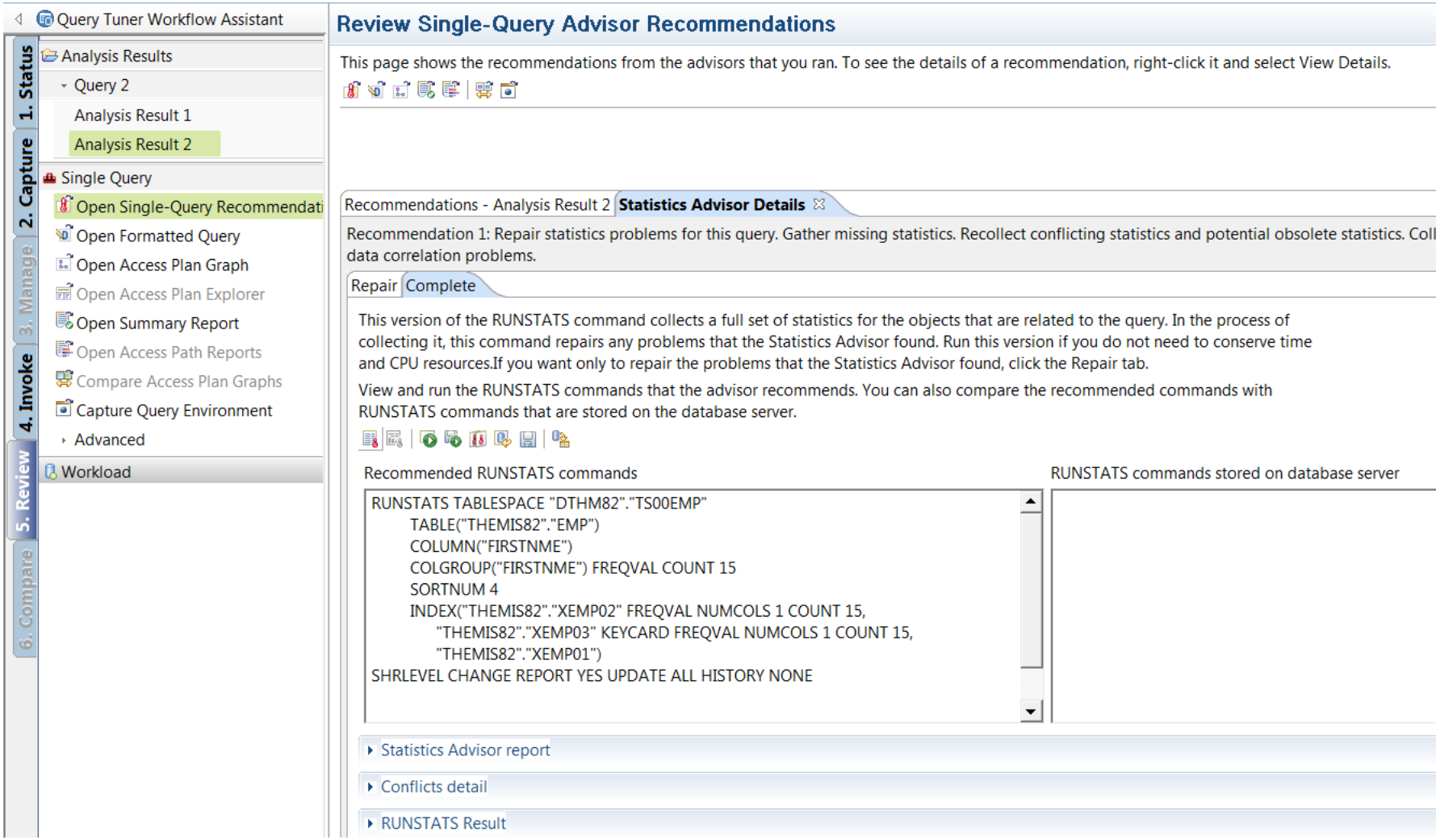


All the output from 'Tune a Query'. Note some are not available in free version.

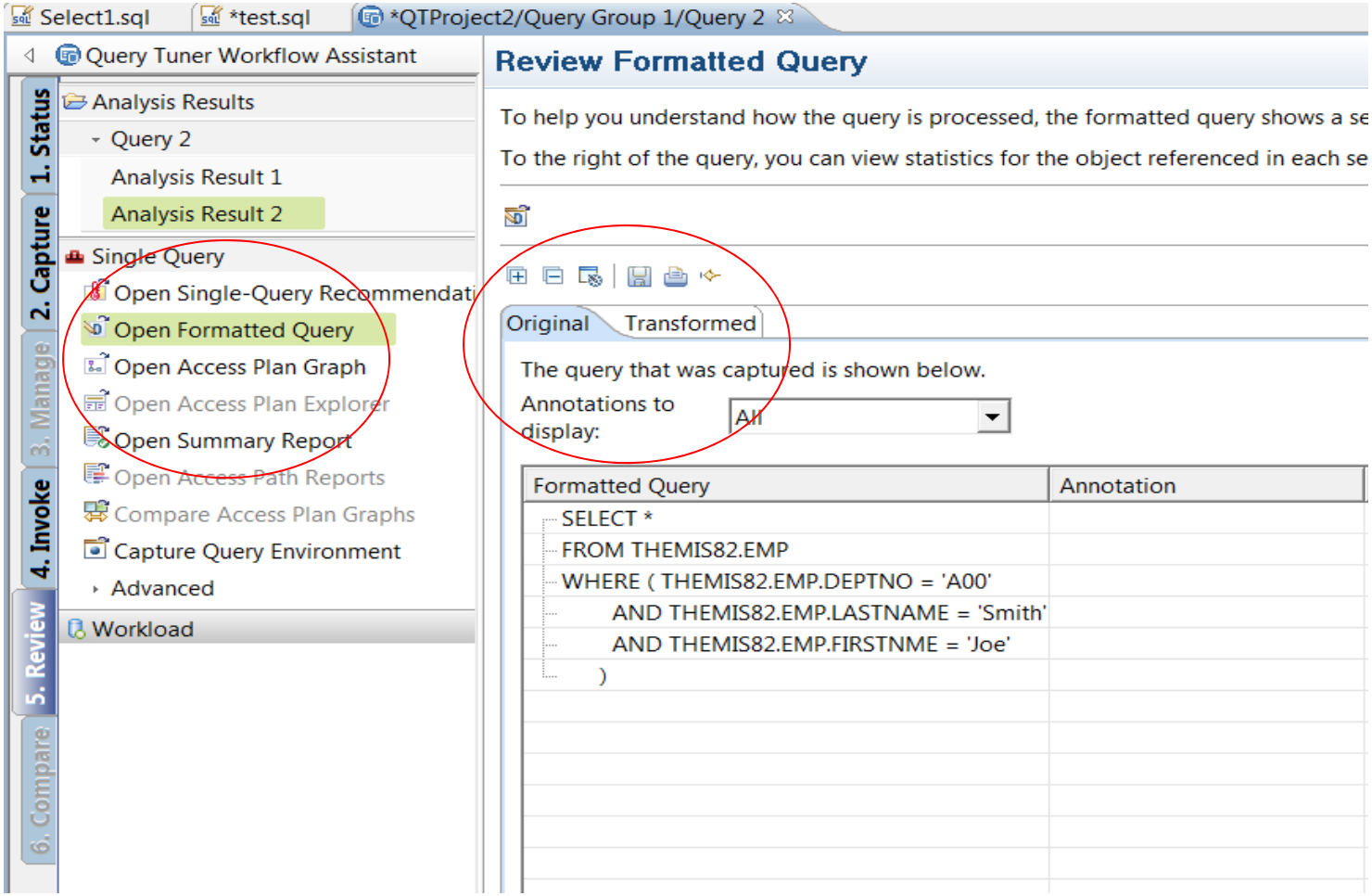
Statistics Advisor



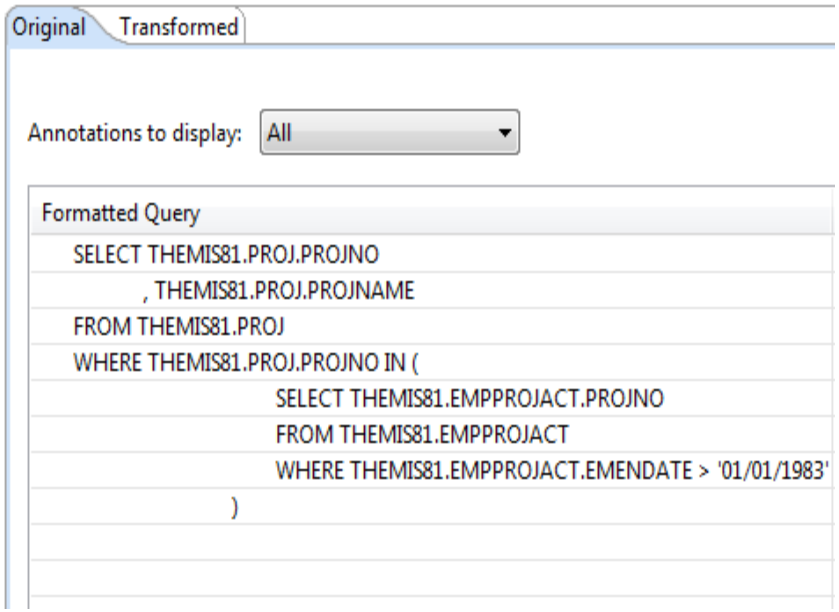
Report


Query Tuner Workflow Assistant
1. Status
2. Capture
3. Manage
4. Invoke
5. Review
6. Compare
Analysis Results
Query 2
Analysis Result 1
Analysis Result 2
Single Query
Open Single-Query Recommendati
Open Formatted Query
Open Access Plan Graph
Open Access Plan Explorer
Open Summary Report
Open Access Path Reports
Compare Access Plan Graphs
Capture Query Environment
Advanced
Workload
Review Single-Query Advisor Recommendations
This page shows the recommendations from the advisors that you ran. To see the details of a recommendation, right-click it and select View Details.
Recommendations - Analysis Result 2
Statistics Advisor Details
Recommendation 1: Repair statistics problems for this query. Gather missing statistics. Recollect conflicting statistics and potential obsolete statistics. Coll
data correlation problems.
Repair
Complete
This version of the RUNSTATS command collects a full set of statistics for the objects that are related to the query. In the process of collecting it, this command repairs any problems that the Statistics Advisor found. Run this version if you do not need to conserve time and CPU resources.If you want only to repair the problems that the Statistics Advisor found, click the Repair tab.
View and run the RUNSTATS commands that the advisor recommends. You can also compare the recommended commands with RUNSTATS commands that are stored on the database server.
Recommended RUNSTATS commands
RUNSTATS TABLESPACE "DTHM82"."TS00EMP"
TABLE("THEMIS82"."EMP")
COLUMN("FIRSTNME")
COLGROUP("FIRSTNME") FREQVAL COUNT 15
SORTNUM 4
INDEX("THEMIS82"."XEMP02" FREQVAL NUMCOLS 1 COUNT 15,
"THEMIS82"."XEMP03" KEYCARD FREQVAL NUMCOLS 1 COUNT 15,
"THEMIS82"."XEMP01")
SHRLEVEL CHANGE REPORT YES UPDATE ALL HISTORY NONE
RUNSTATS commands stored on database server
Statistics Advisor report
Conflicts detail
RUNSTATS Result

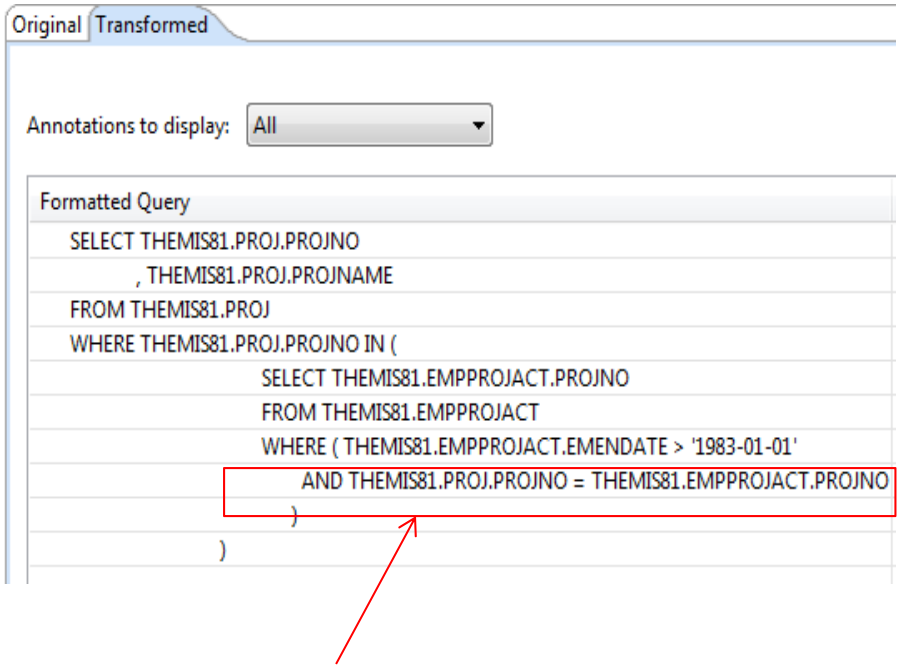
z/OS Tune a query – Query Transformation



z/OSTune a query – Query Transformation

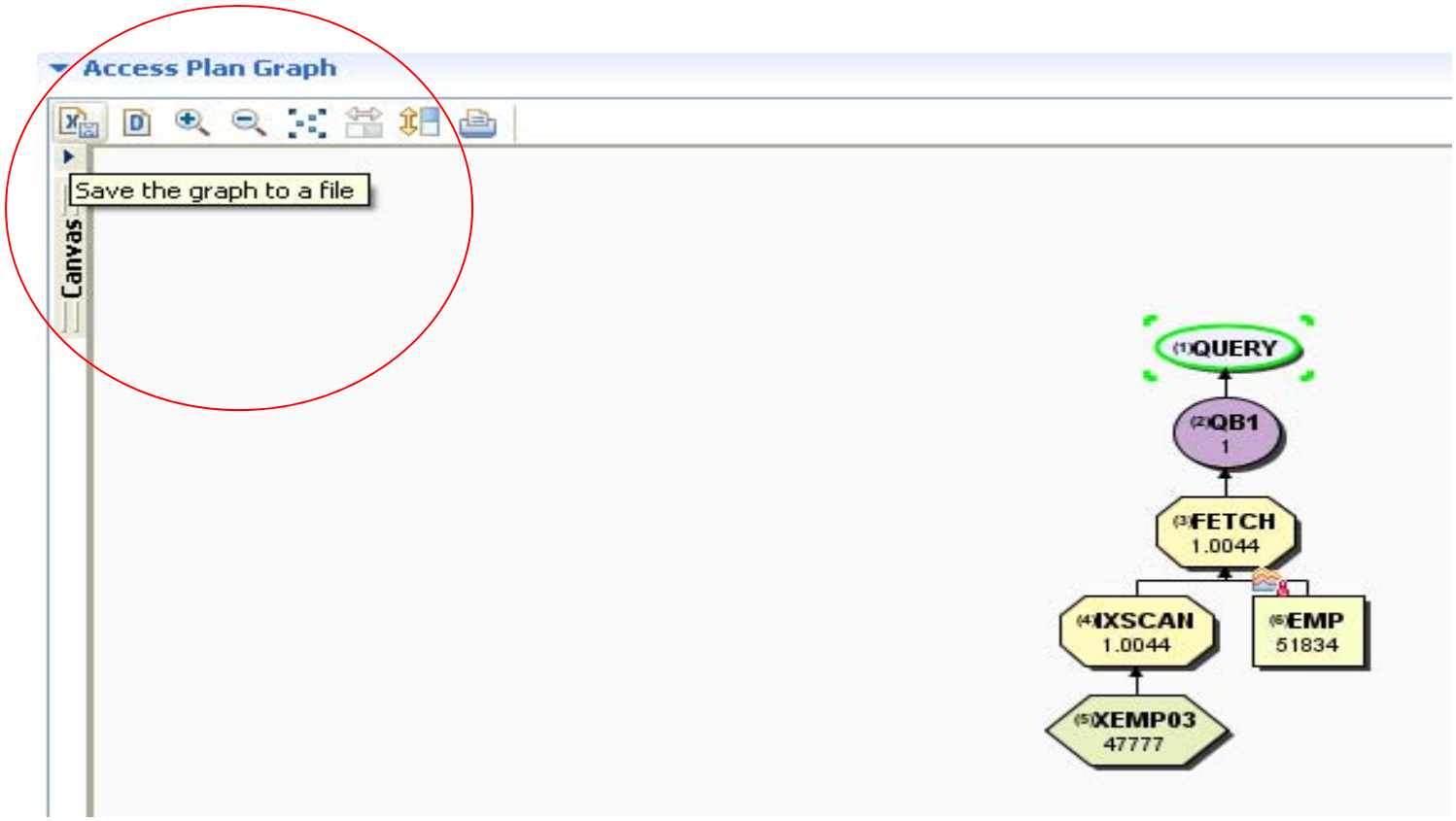


Note:
Non Correlated

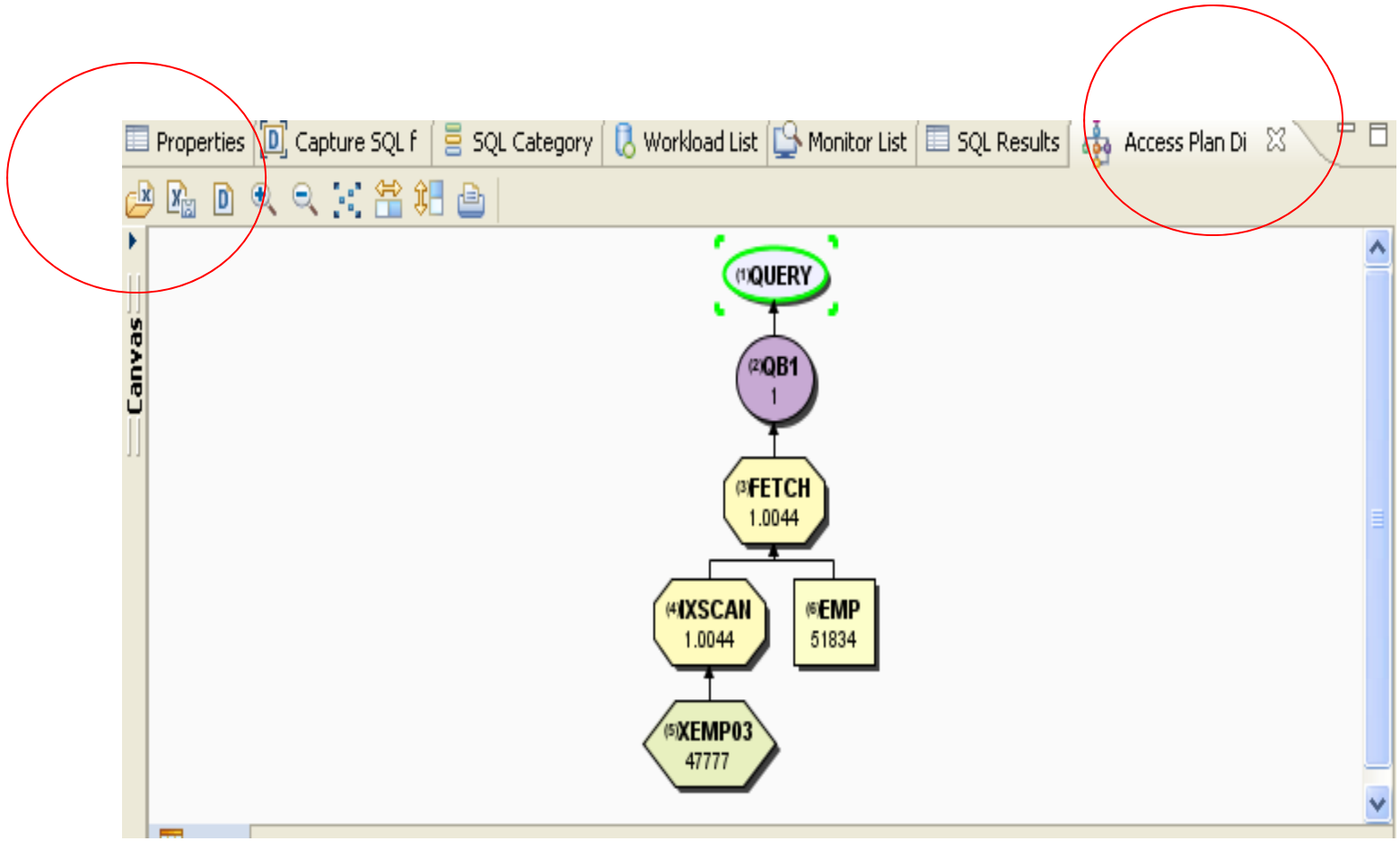


Note:
Correlated

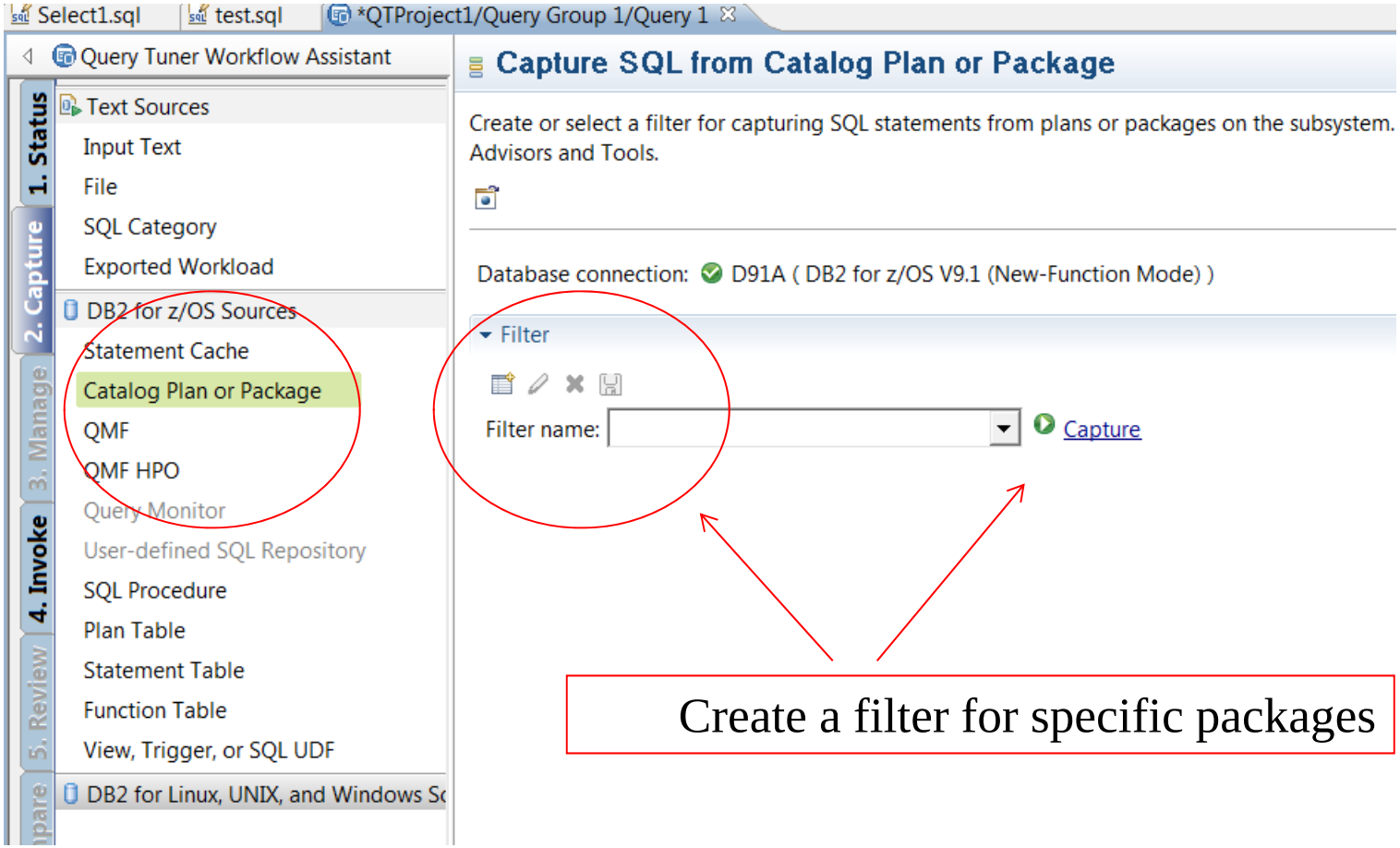
Saving off an Access Path



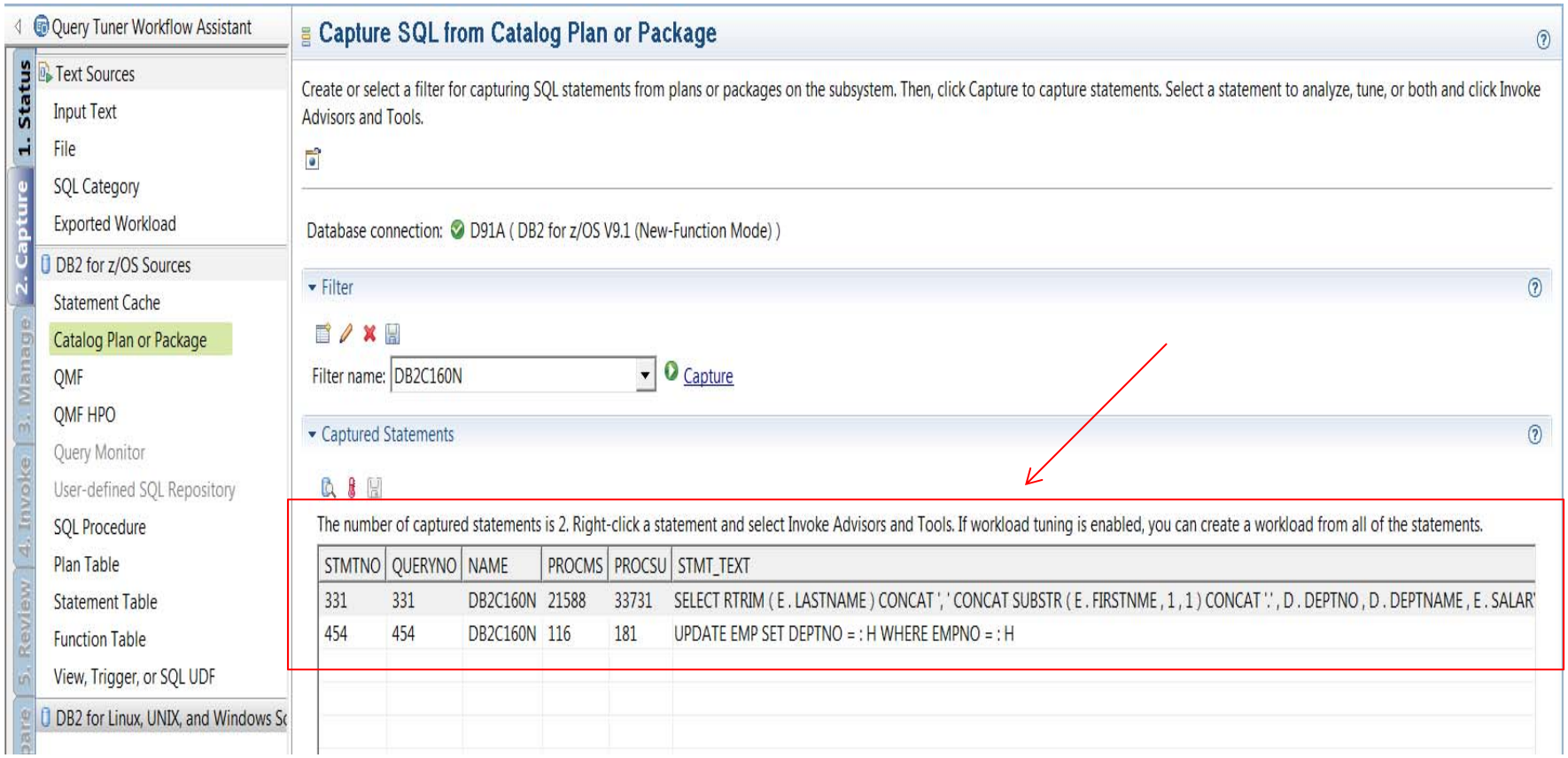
Opening an Access Path



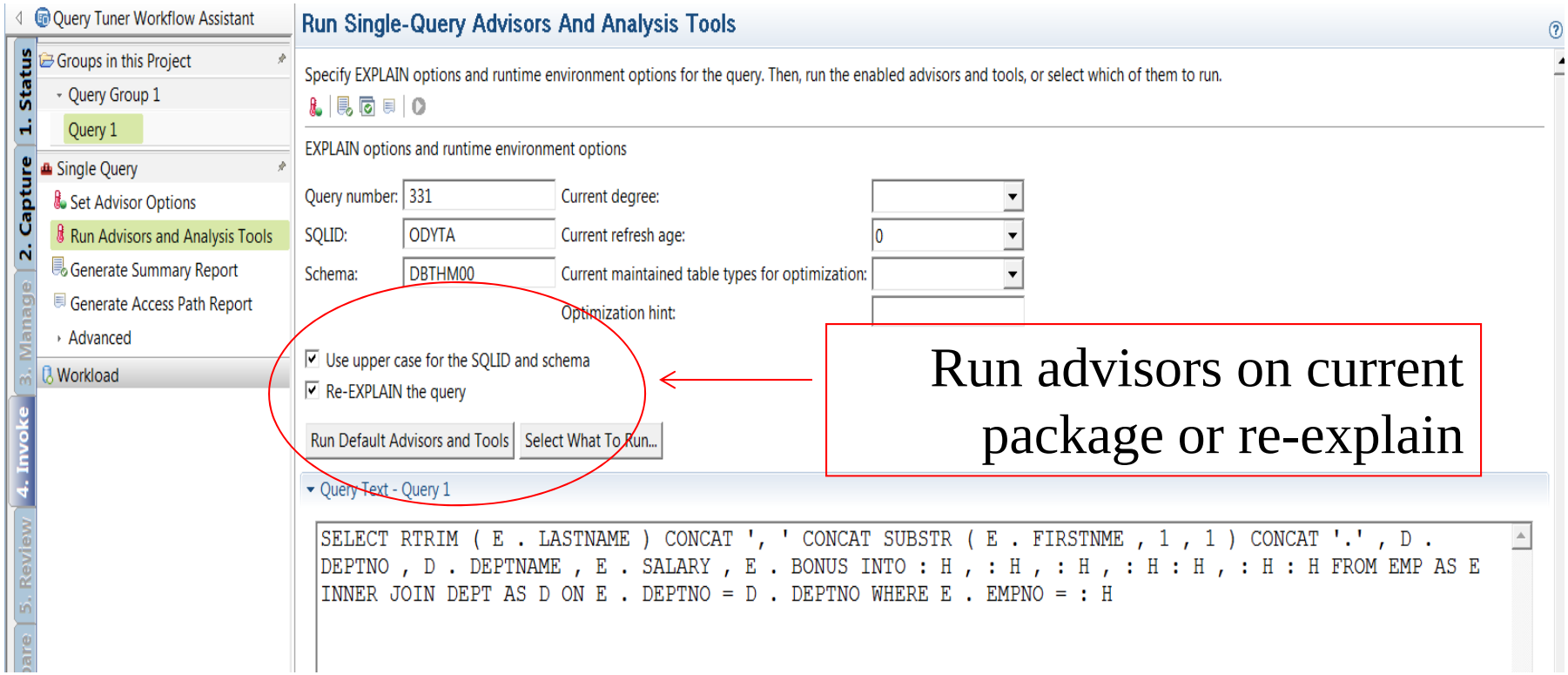
Retrieving Queries From Packages



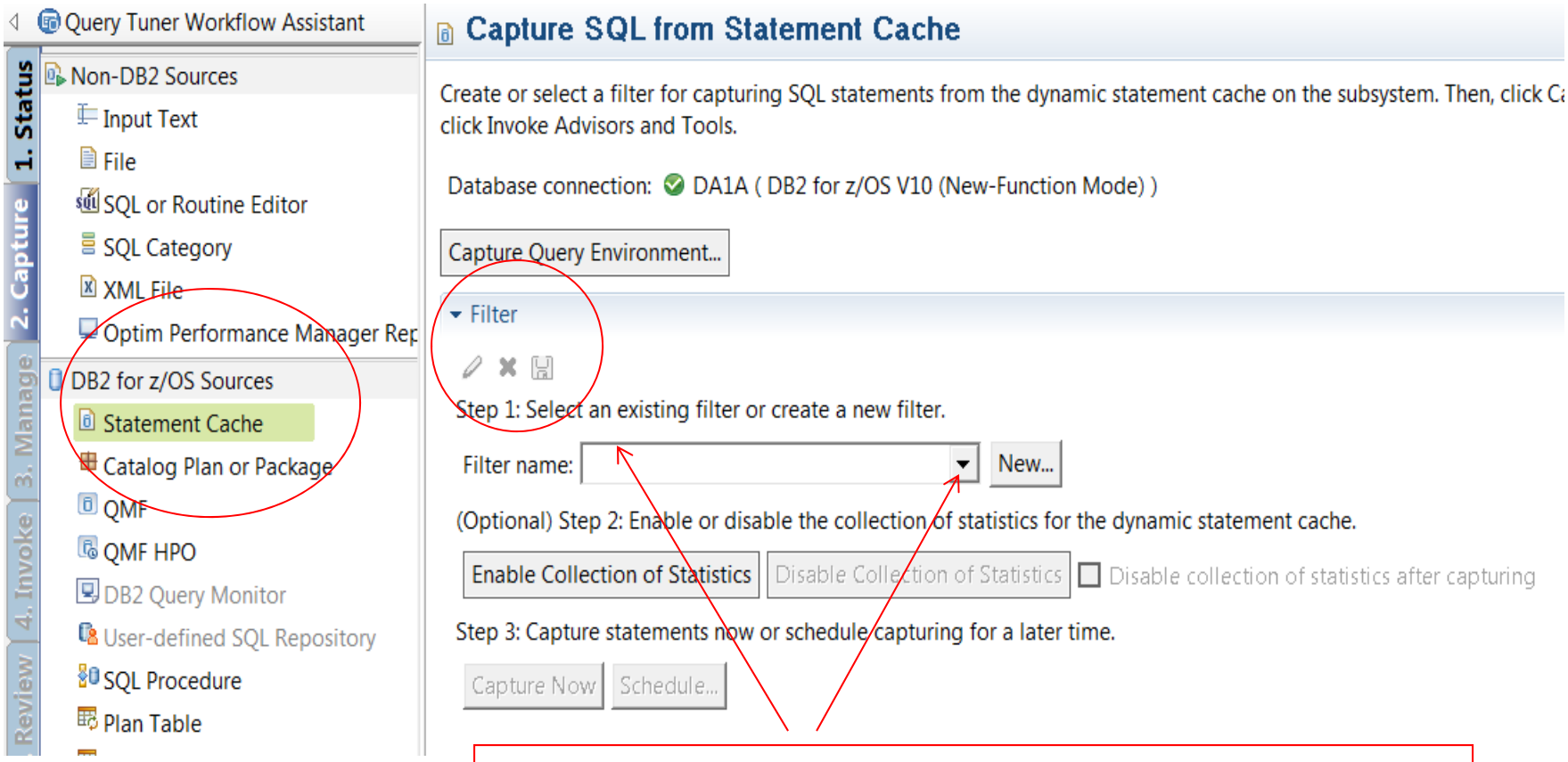
Retrieving Queries From Packages



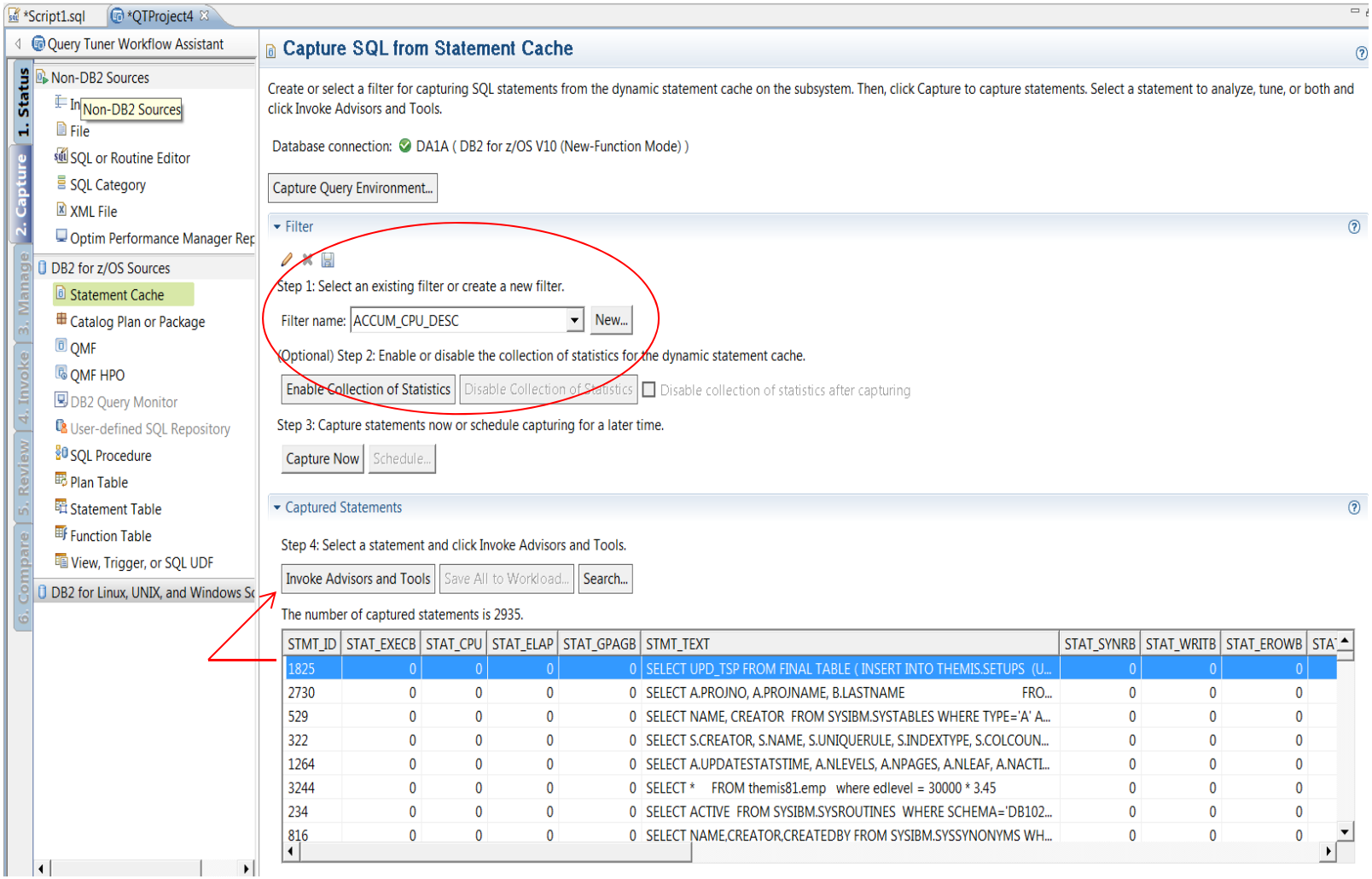
Retrieving Queries From Packages



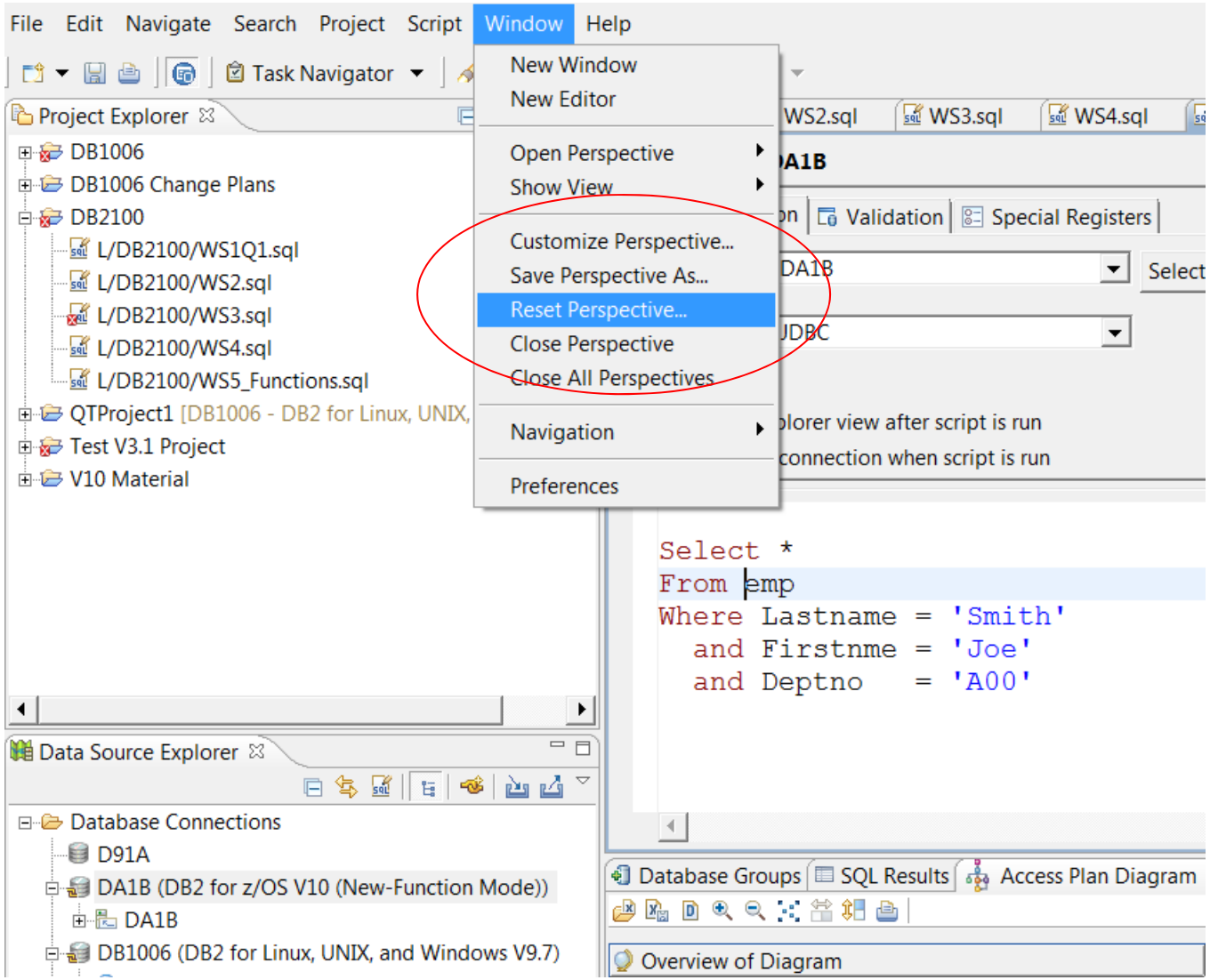
Retrieving Queries From Statement Cache



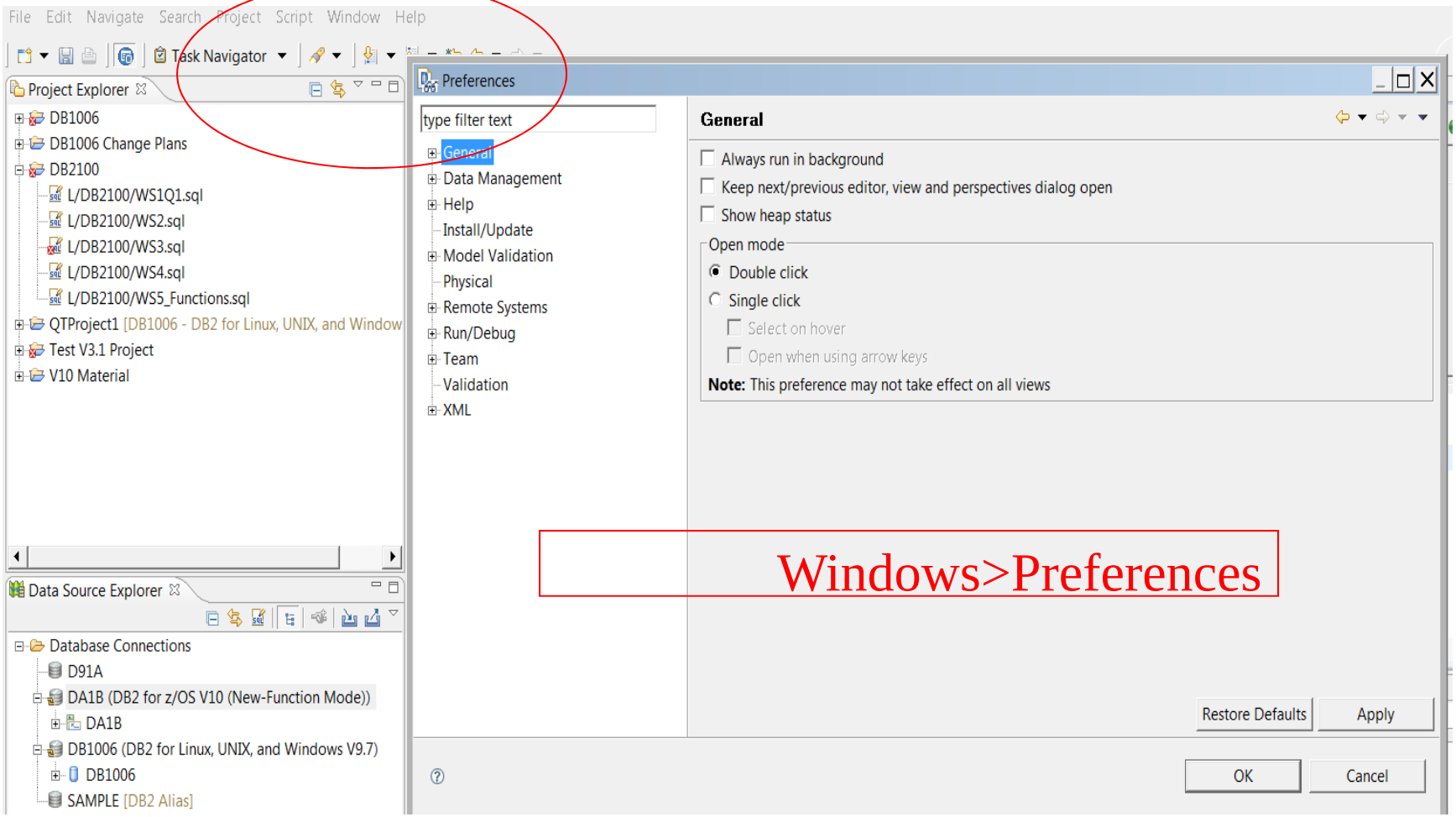
Retrieving Queries From Statement Cache

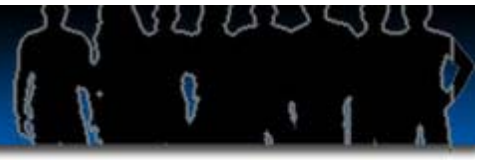


Everyone's Most Favorite Option



Preference Settings





Query Tuning Documents

Search on:

Tuning SQL With Optim Query Tuner

Part 1 and Part 2

Thank You for Attending CODUG!

"There is always time for an Explain"

"I have noticed that when the developers get educated, good SQL programming standards are in place, program walkthroughs and Explains are executed correctly, incident reporting stays low, CPU costs do not get out of control, and most performance issues are found before promoting code to production."