

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**



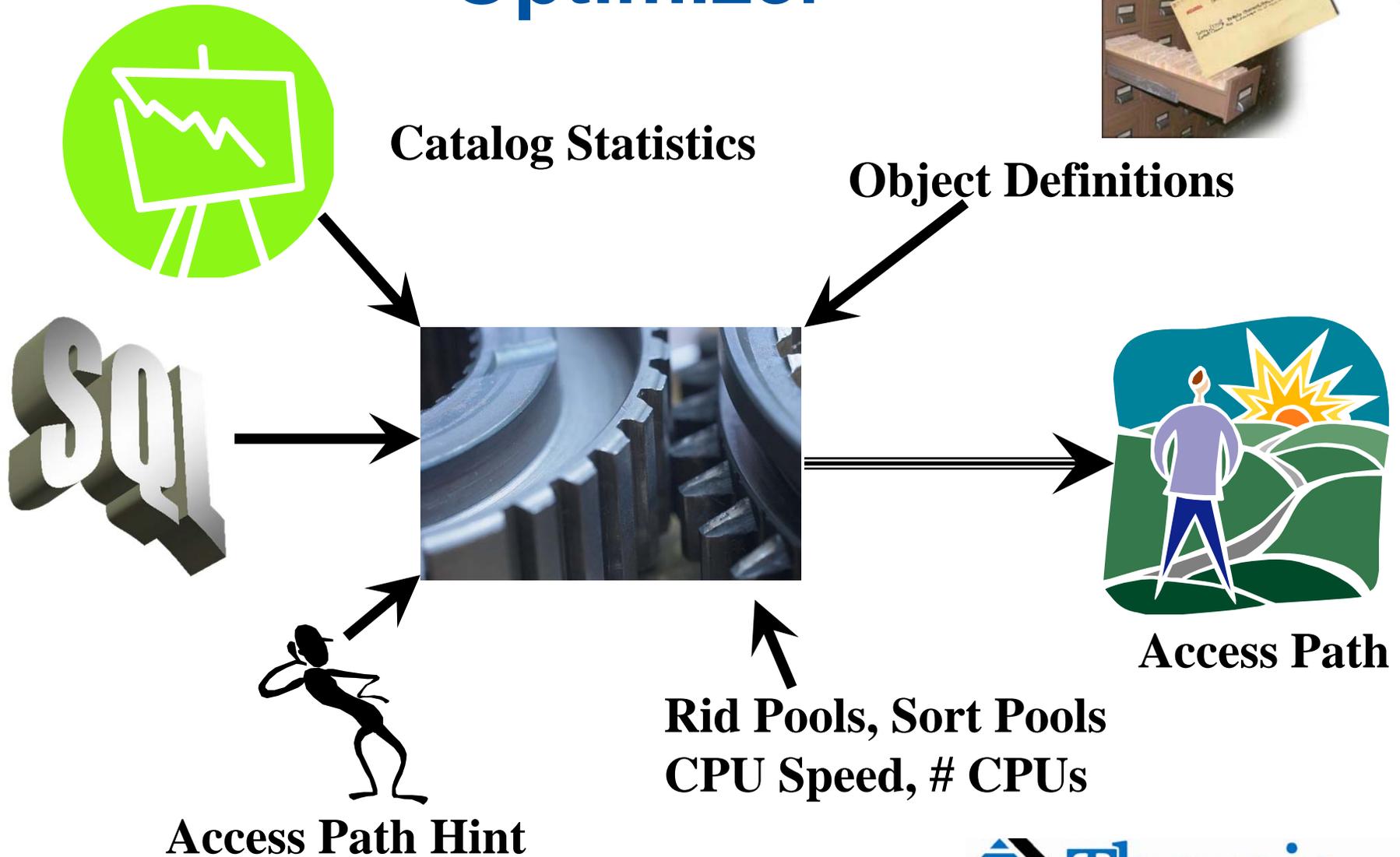


Developers Should Focus On

- **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**



Optimizer



Explain

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

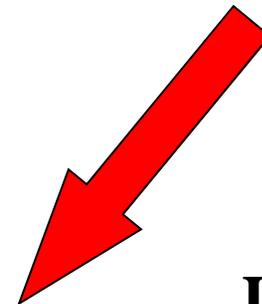
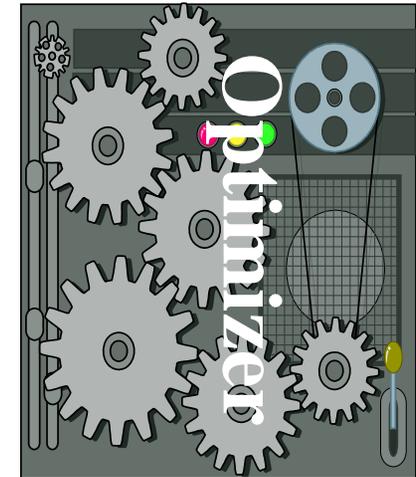


OR

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

z/OS

**PLAN_TABLE
DSN_STATEMNT_TABLE
DSN_FUNCTION_TABLE
& a bunch of “other” tables**

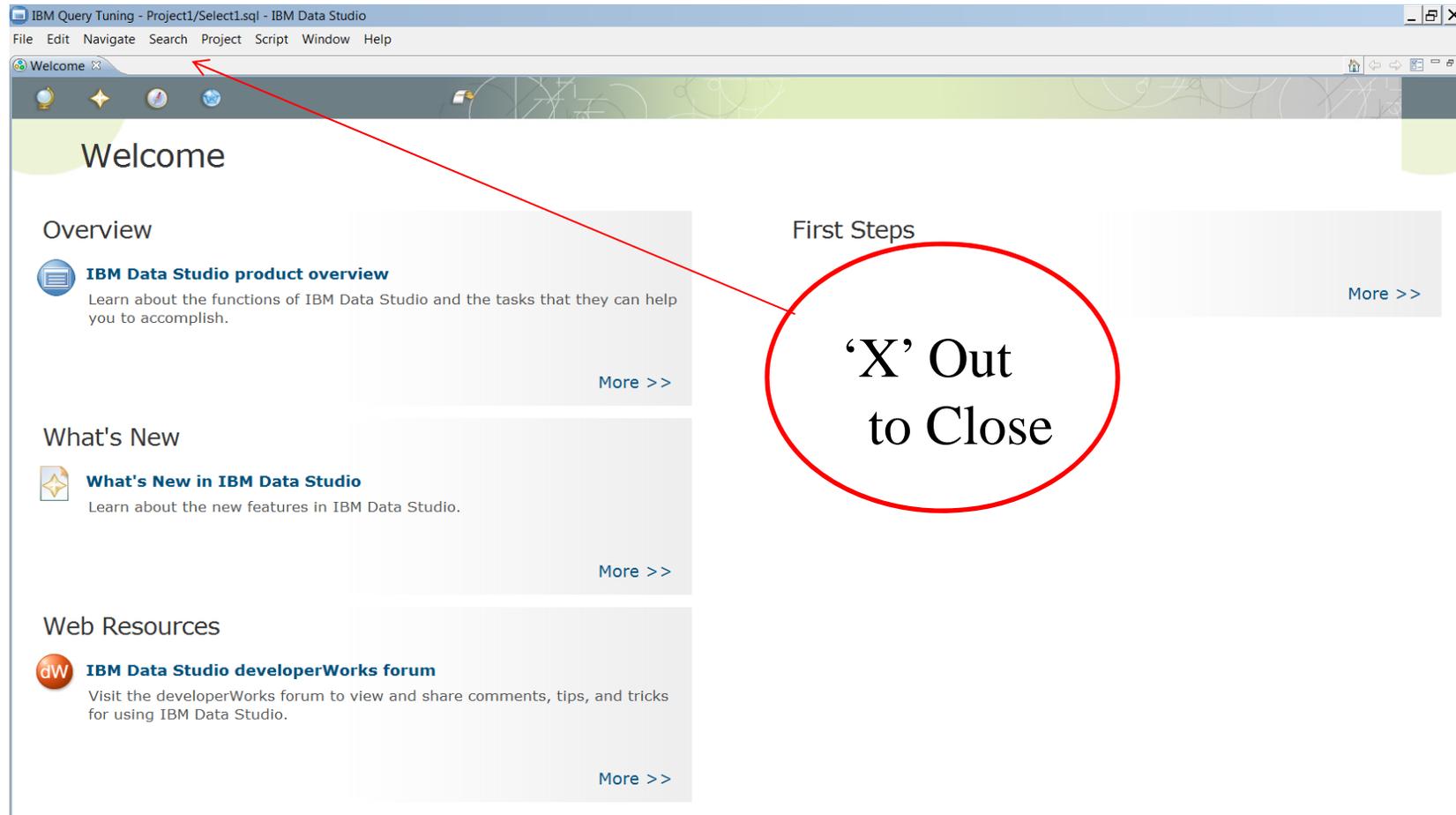


LUW

**EXPLAIN_STATEMENT
EXPLAIN_PREDICATE
& a bunch of “other” tables**



IBM Data Studio



IBM Query Tuning - Project1/Select1.sql - IBM Data Studio

File Edit Navigate Search Project Script Window Help

Welcome

Welcome

Overview

 **IBM Data Studio product overview**
Learn about the functions of IBM Data Studio and the tasks that they can help you to accomplish.

[More >>](#)

What's New

 **What's New in IBM Data Studio**
Learn about the new features in IBM Data Studio.

[More >>](#)

Web Resources

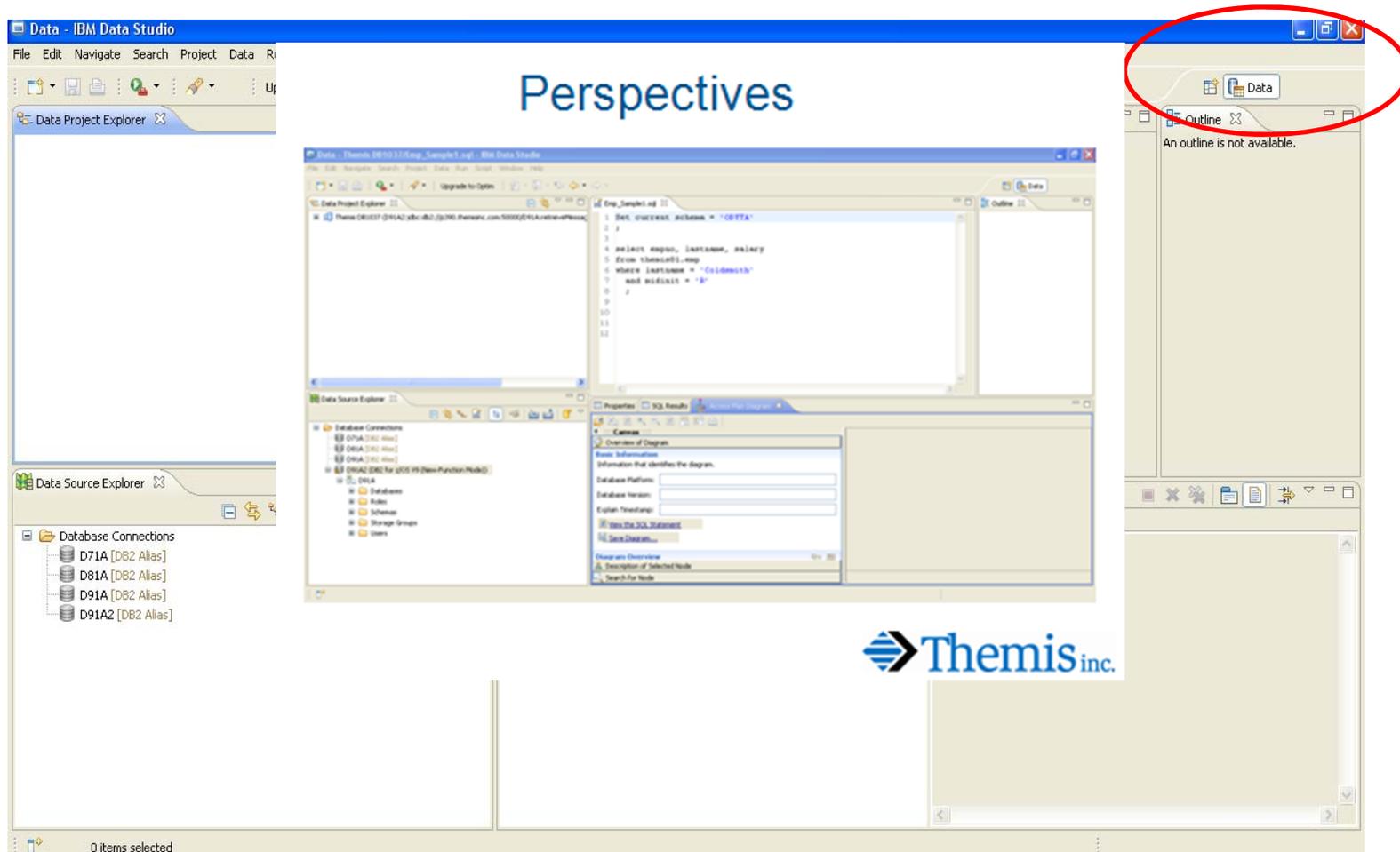
 **IBM Data Studio developerWorks forum**
Visit the developerWorks forum to view and share comments, tips, and tricks for using IBM Data Studio.

[More >>](#)

First Steps [More >>](#)

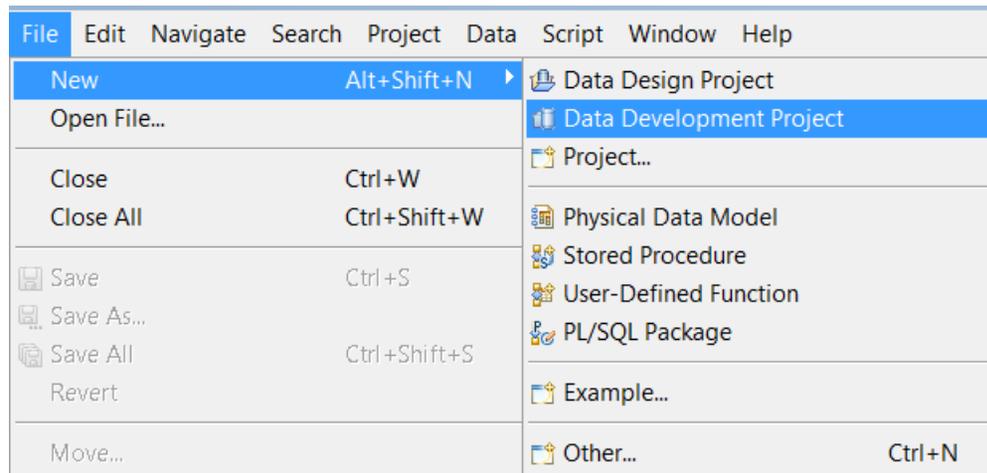
'X' Out to Close

IBM Data Studio





Create a Project

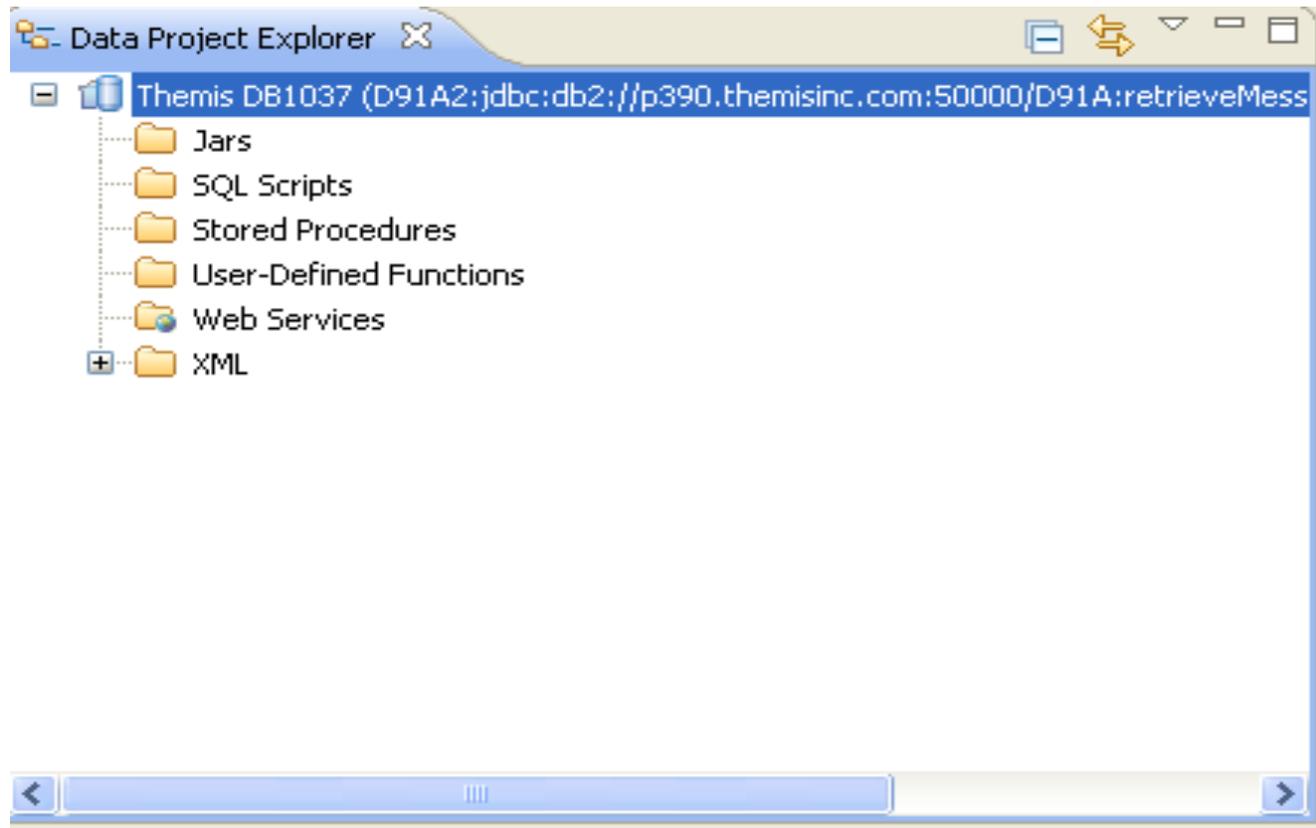


When Creating a Project

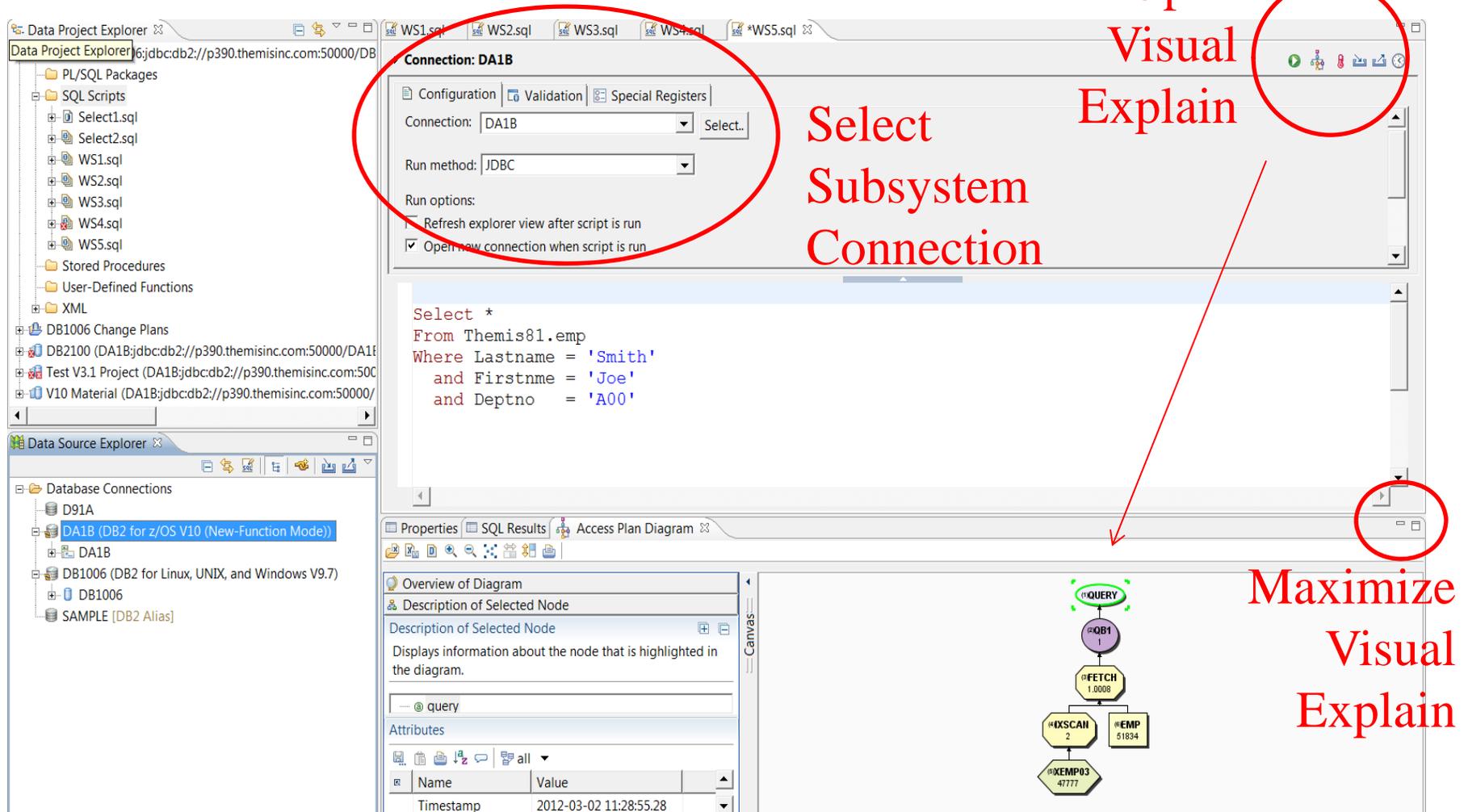
- 1) Assign a Project Name and Type
(Data Devlp Project)
- 1) Assign a Subsystem
- 2) Assign a Default Schema



Data Studio Project Created



Data Studio Explaining Queries



The screenshot shows the IBM Data Studio interface with several annotations in red:

- Select Subsystem Connection:** A red circle highlights the 'Connection: DA1B' dropdown menu in the 'Configuration' tab of the 'Connection: DA1B' panel.
- Open Visual Explain:** A red circle highlights the 'Visual Explain' icon in the toolbar.
- Maximize Visual Explain:** A red circle highlights the maximize icon in the bottom right corner of the 'Access Plan Diagram' window.

The central query editor contains the following SQL:

```
Select *  
From Themis81.emp  
Where Lastname = 'Smith'  
and Firstname = 'Joe'  
and Deptno = 'A00'
```

The 'Access Plan Diagram' window displays the following execution plan:

```
graph TD  
  Q[QUERY] --> QB1[QB1 1]  
  QB1 --> F[FETCH 1.0008]  
  F --> X[XSSCAN 2]  
  F --> E[EMP 51834]  
  X --> XEMP03[XEMP03 4777]
```

The 'Overview of Diagram' panel shows the description of the selected node:

Description of Selected Node
Displays information about the node that is highlighted in the diagram.

Attributes

Name	Value
Timestamp	2012-03-02 11:28:55.28

z/OS Data Studio Access Path Graphs

The screenshot displays the z/OS Data Studio interface. On the left, the 'Canvas' pane shows the 'Description of Selected Node' and 'Attributes' sections, both circled in red. The 'Attributes' section includes a table with the following data:

Name	Value
Type	SELECT
CPU Cost (ms)	1
CPU Cost (su)	2
Cost Category	A
Reason	

The main 'Access Plan Diagram' on the right shows a hierarchical flow of operations:

- (1) QUERY (green oval, highlighted)
- (2) QB1 (purple circle, 1)
- (3) FETCH (yellow octagon, 1.0044)
- (4) IXSCAN (yellow octagon, 2) and (6) EMP (yellow rectangle, 51834) are inputs to the FETCH node.
- (5) XEMP03 (yellow hexagon, 47777) is an input to the IXSCAN node.

LUW Data Studio Access Path Graphs

The screenshot displays the LUW Data Studio interface with an Access Plan Diagram. The diagram shows a hierarchical flow of operations:

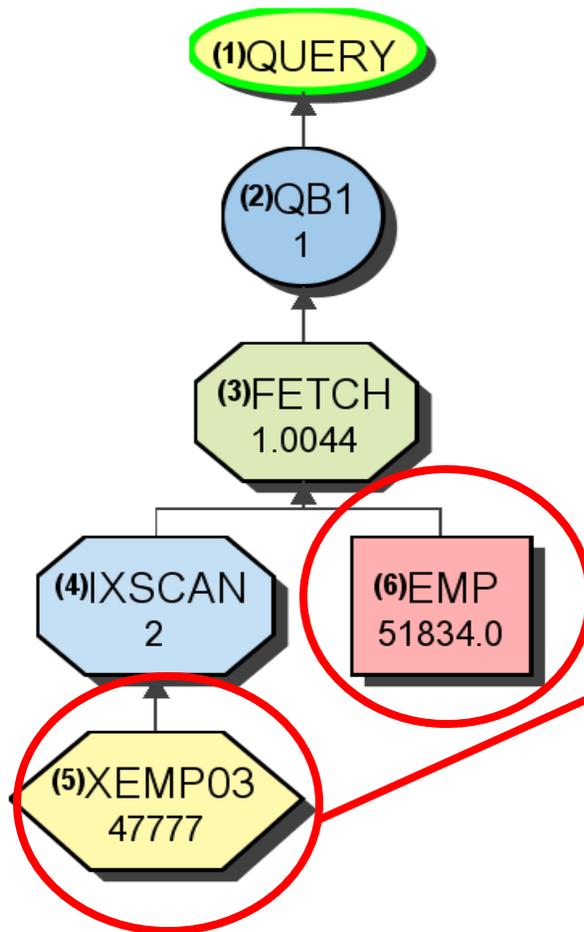
- (1)RETURN** (Circle): Final operation with a cumulative total cost of 22.7952.
- (2)FETCH** (Hexagon): Feeds into the RETURN node with a cumulative total cost of 22.7952.
- (3)IXSCAN** (Hexagon): Feeds into the FETCH node with a value of 15.1305.
- (00)EMP** (Rectangle): Feeds into the FETCH node, representing the table THEMIS81.
- (01)XEMP03** (Hexagon): Feeds into the IXSCAN node, representing the index THEMIS81.

The left-hand pane shows the 'Attributes' table for the selected node:

NAME	VALUE
Operator Identifier	1
Operator Type	RETURN
Estimated Output Cardinality	0.00961296
Cumulative Total Cost	22.7952
Cumulative CPU Cost	89771.6
Cumulative IO cost	3.01356
Cumulative Re-execution Total Cost	0.00214458
Cumulative Re-execution CPU Cost	15134.3



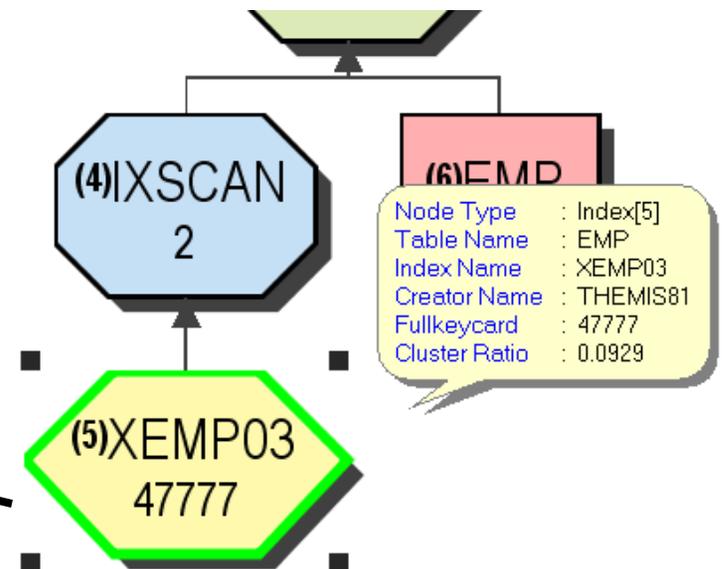
Data Studio Access Path Graphs



Sources of Data

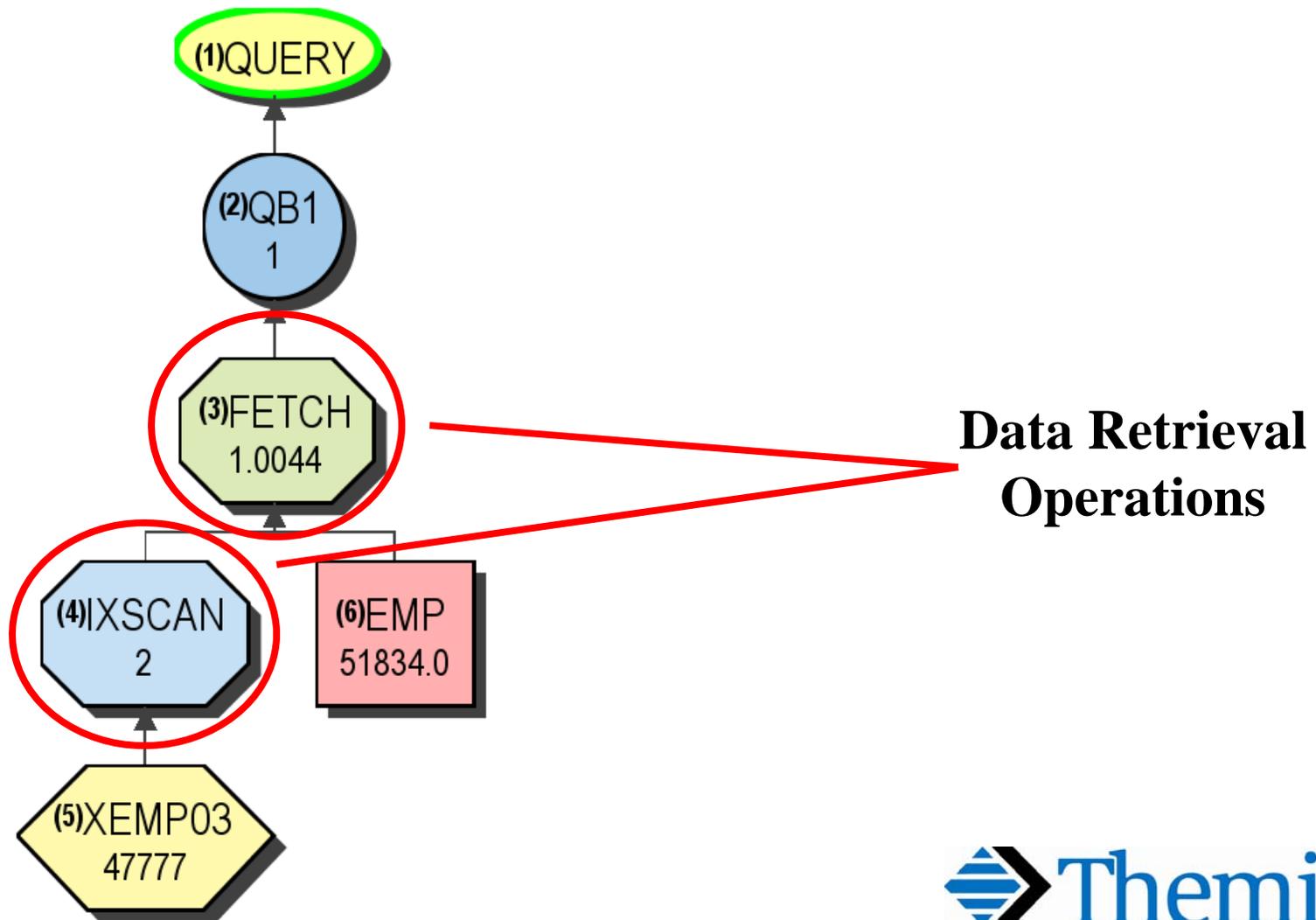
Data Studio Access Path Graphs

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





Data Studio Access Path Graphs



z/OS Stage 1 / 2 Predicates

Overview of Diagram

Description of Selected Node

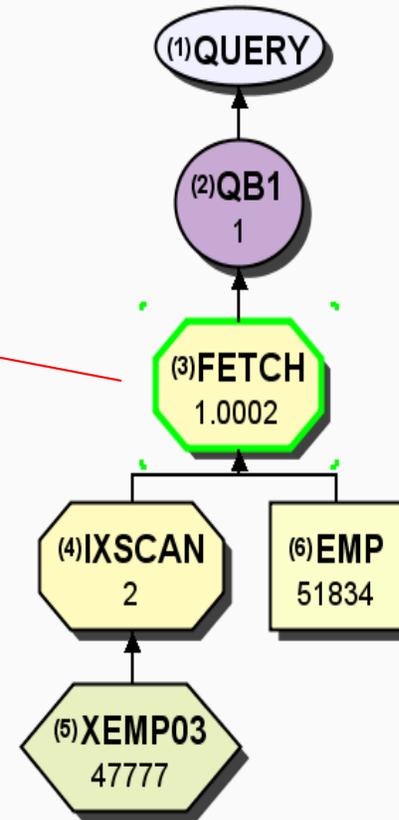
Description of Selected Node

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

- fetch
 - Stage1_Predicates
 - THEMIS81.EMP.DEPTNO='A00'
 - Stage2_Predicates
 - YEAR(THEMIS81.EMP.HIREDATE)=2001

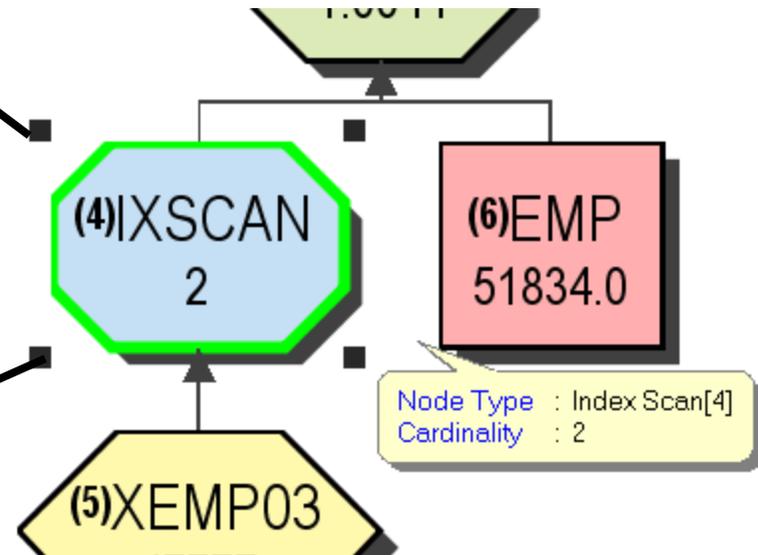
Attributes

Name	Value
Input Cardinality	2
Scanned Rows	2
Stage 1 Predicates	Filter Factor
THEMIS81.EMP.DEPTNO='A00'	0.0092
Stage 1 Returned Rows	1.0044
Stage 2 Predicates	Filter Factor
YEAR(THEMIS81.EMP.HIREDATE)=2001	0.04
Stage 2 Returned Rows	1.0002
Outout Cardinality	1.0002



Data Studio Access Path Graphs

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



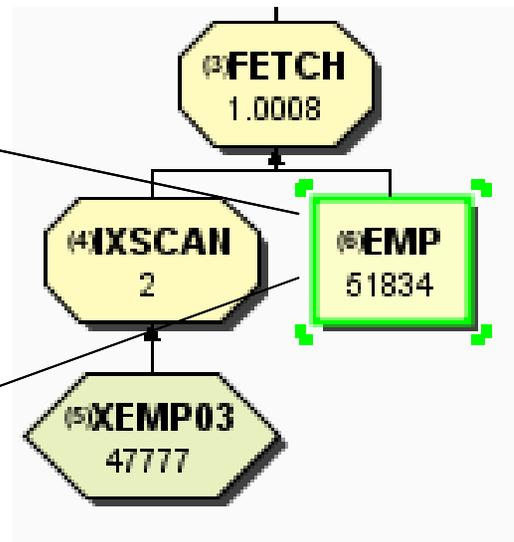
Data Studio Access Path Graphs

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

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

Attributes

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

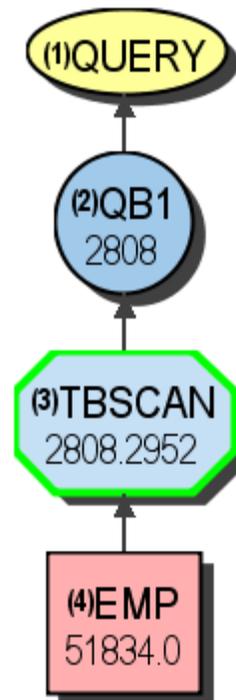


Tablespace Scan

```
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

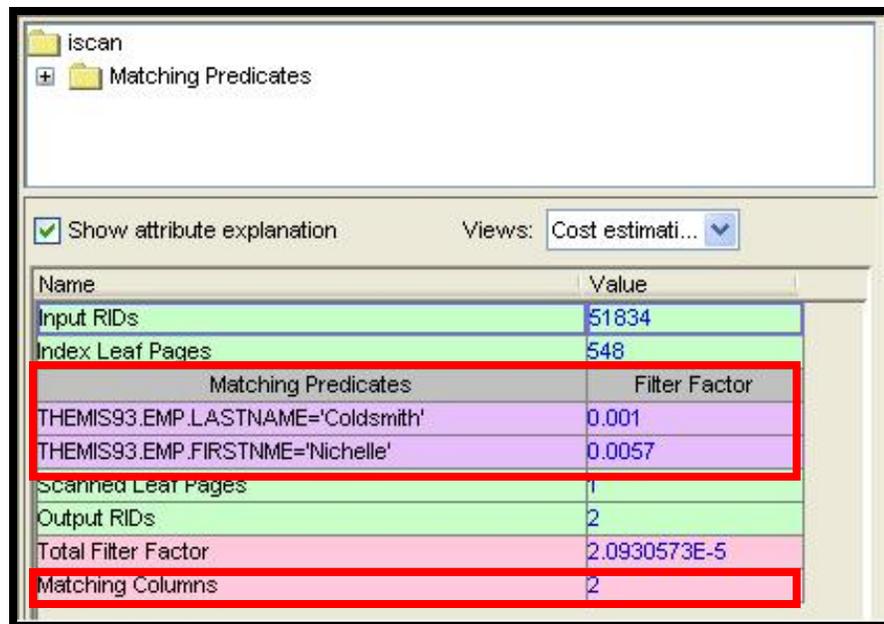


z/OS Index Scan - Matching

```
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	

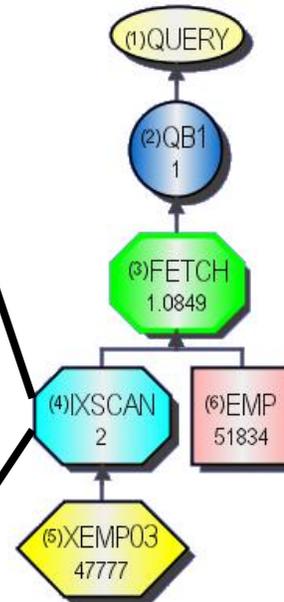


iscan

- Matching Predicates

Show attribute explanation Views: Cost estimati...

Name	Value
Input RIDs	51834
Index Leaf Pages	548
Matching Predicates	
Filter Factor	
THEMIS93.EMP.LASTNAME='Coldsmith'	0.001
THEMIS93.EMP.FIRSTNME='Nichelle'	0.0057
Scanned Leaf Pages	1
Output RIDs	2
Total Filter Factor	2.0930573E-5
Matching Columns	2





z/OS Index Screening

INDEX XEMP03 on
(LASTNAME, FIRSTNAME, 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	

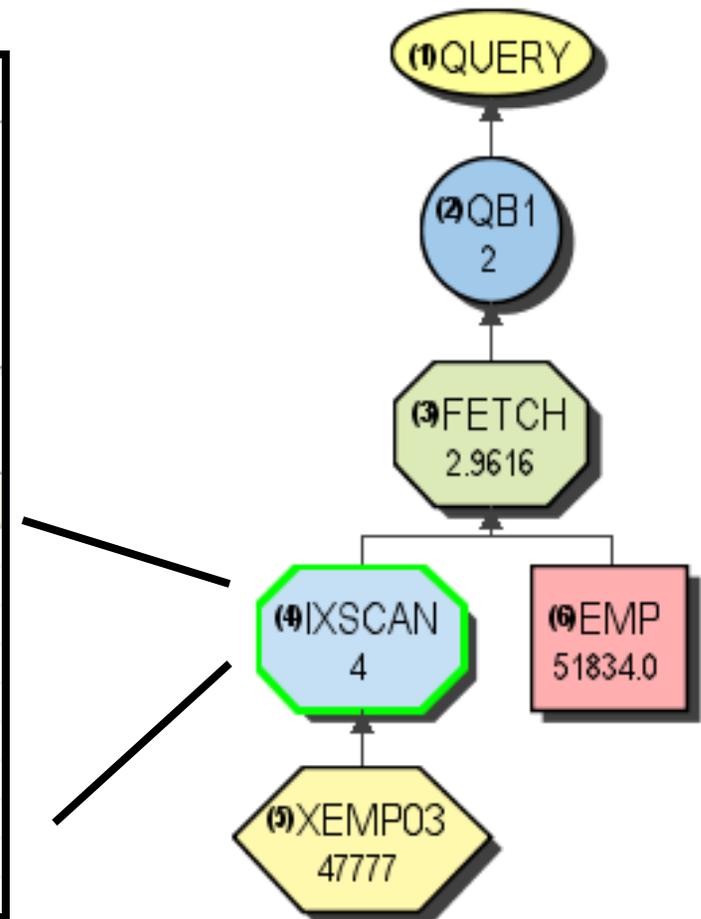
z/OS Index Screening (cont)

Index Scan: IXSCAN

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

Show attribute explanation Views: cost_estimal

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'
```

Overview of Diagram

Description of Selected Node

Description of Selected Node

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

ixscan

- Stream
- Predicates
 - predicate
 - predicate

Attributes

NAME	VALUE
Predicate identifier	2
How the predicate is applied	Stop Key Predicate
When the predicate is applied	
Relational operation type	Less Than or Equal
Subquery	No
Filter factor	0.0528388
Predicate text	(Q1.EMPNO <= '099999')

Description of the Selected Attribute

Indicates when the subquery used in this predicate is evaluated

Search for Node

Query

```

graph BT
    XEMP01["(01)XEMP01  
THEMIS80"] --> IXSCAN["(5)IXSCAN  
95.5216"]
    IXSCAN --> SORT["(4)SORT  
96.4385"]
    SORT --> RIDSCN["(3)RIDSCN  
96.4389"]
    EMP["(00)EMP  
THEMIS80"] --> RIDSCN
    RIDSCN --> FETCH["(2)FETCH  
169.805"]
    FETCH --> RETURN["(1)RETURN  
169.805"]
    
```



z/OS Index Scan - Nonmatching

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

PLAN_TABLE

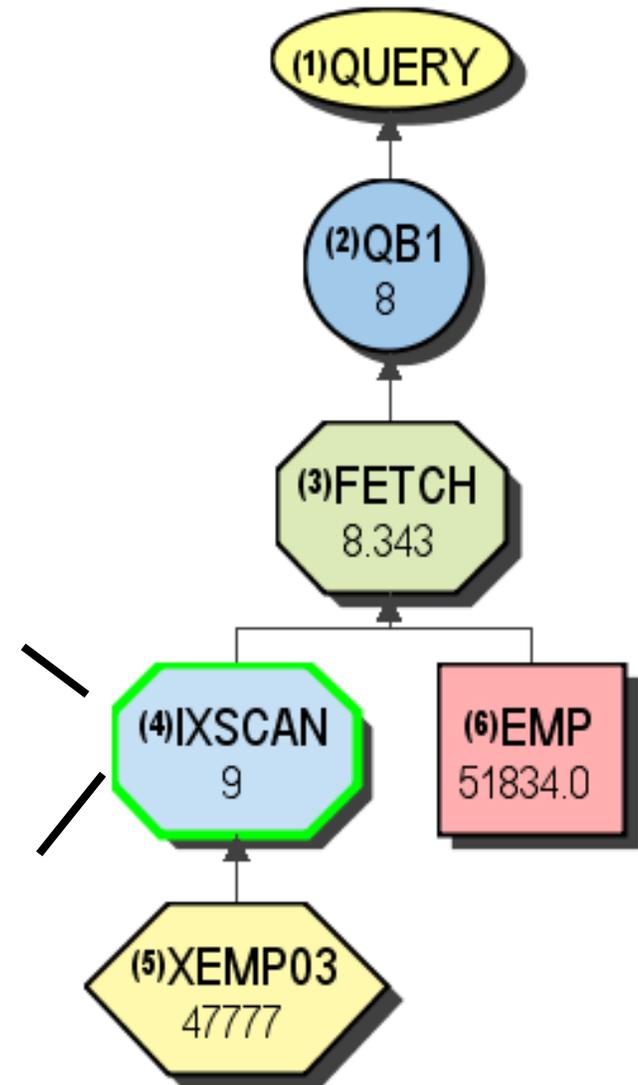
PLAN NO	METHOD	TNAME	ACCESS TYPE	MATCH COLS	ACCESS NAME	INDEX ONLY	PREFETCH
1	0	EMP	I	0	XEMP03	N	



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'
```

Overview of Diagram

Description of Selected Node

Description of Selected Node

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

- ixscan
 - Stream
 - Predicates
 - @ predicate
 - @ predicate

Attributes

NAME	VALUE
Predicate identifier	3
How the predicate is applied	Sargable Predicate
When the predicate is applied	
Relational operation type	Equal
Subquery	No
Filter factor	0.00415426
Predicate text	(Q1.FIRSTNME = 'David')

Description of the Selected Attribute

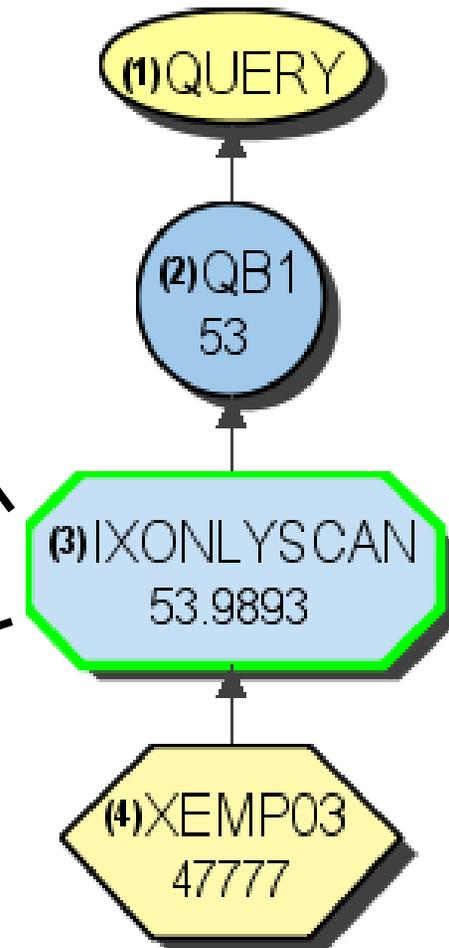
Indicates when the subquery used in this predicate is evaluated

```

    graph BT
      A["(00)XEMP03  
THEMIS80"] --> B["(2)IXSCAN  
726.287"]
      B --> C["(1)RETURN  
726.287"]
  
```

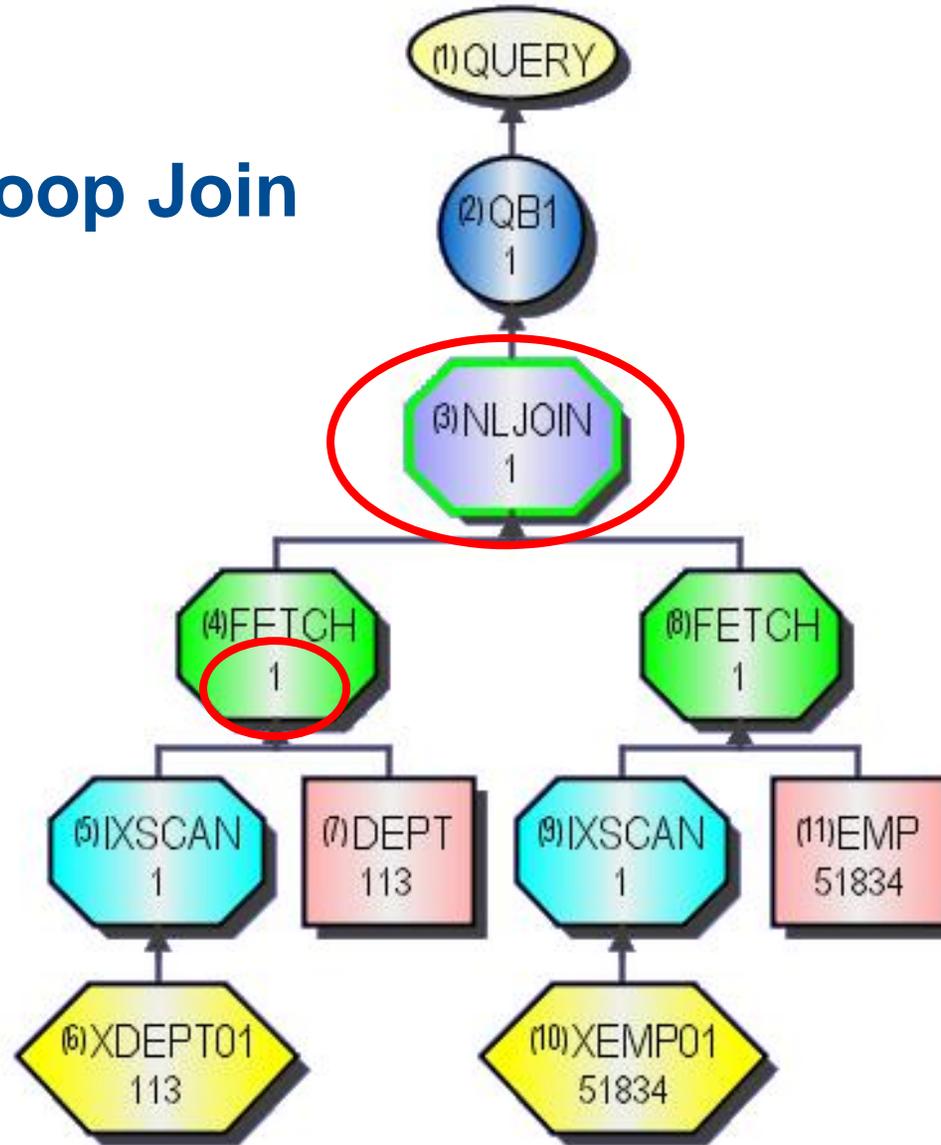
Index Only Access

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



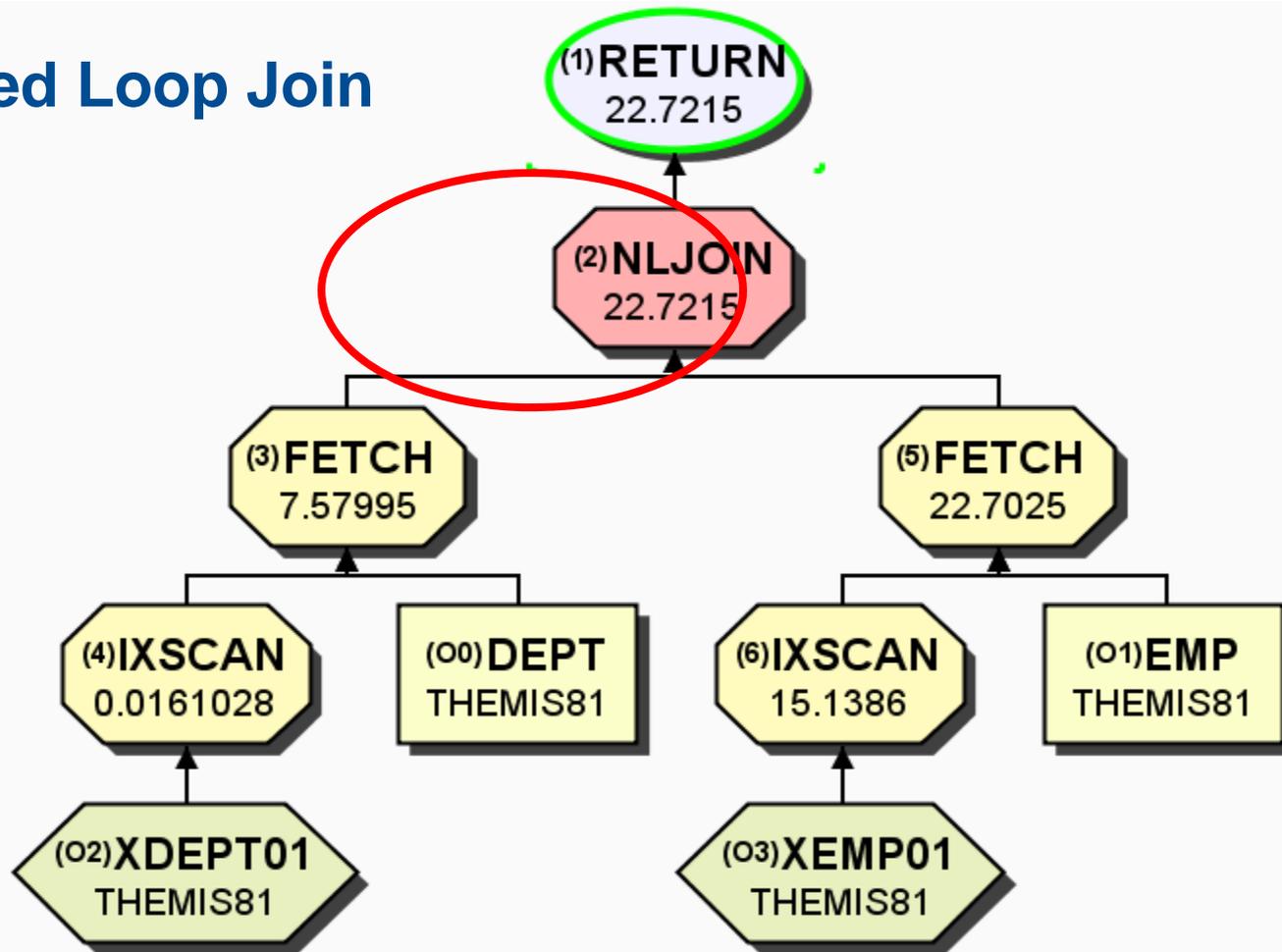
SELECT LASTNAME, FIRSTNME, MIDINIT
FROM EMP
WHERE LASTNAME LIKE 'Jo%'

z/OS Nested Loop Join

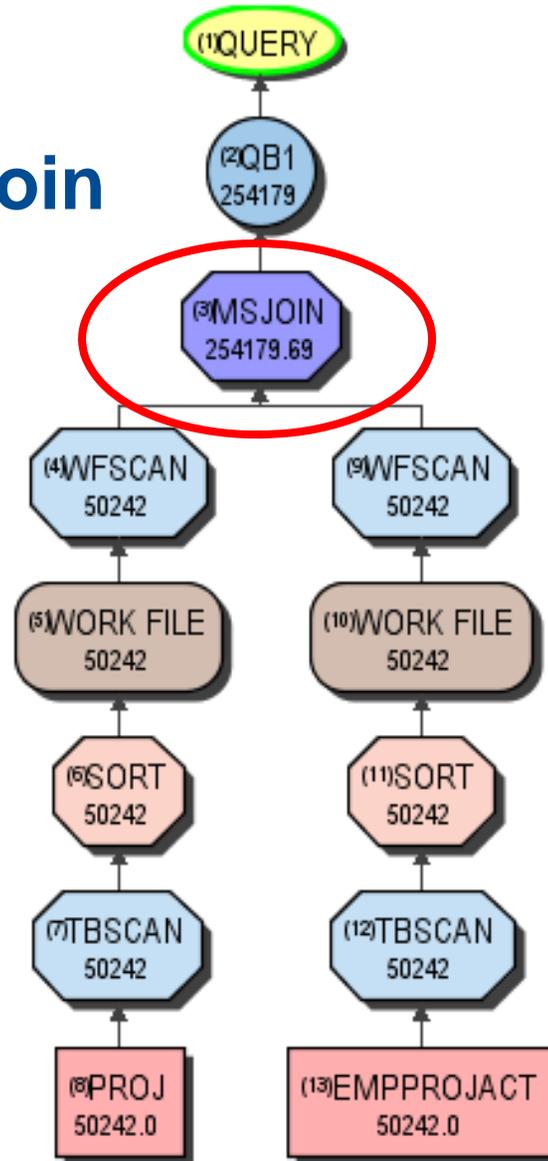




LUW Nested Loop Join

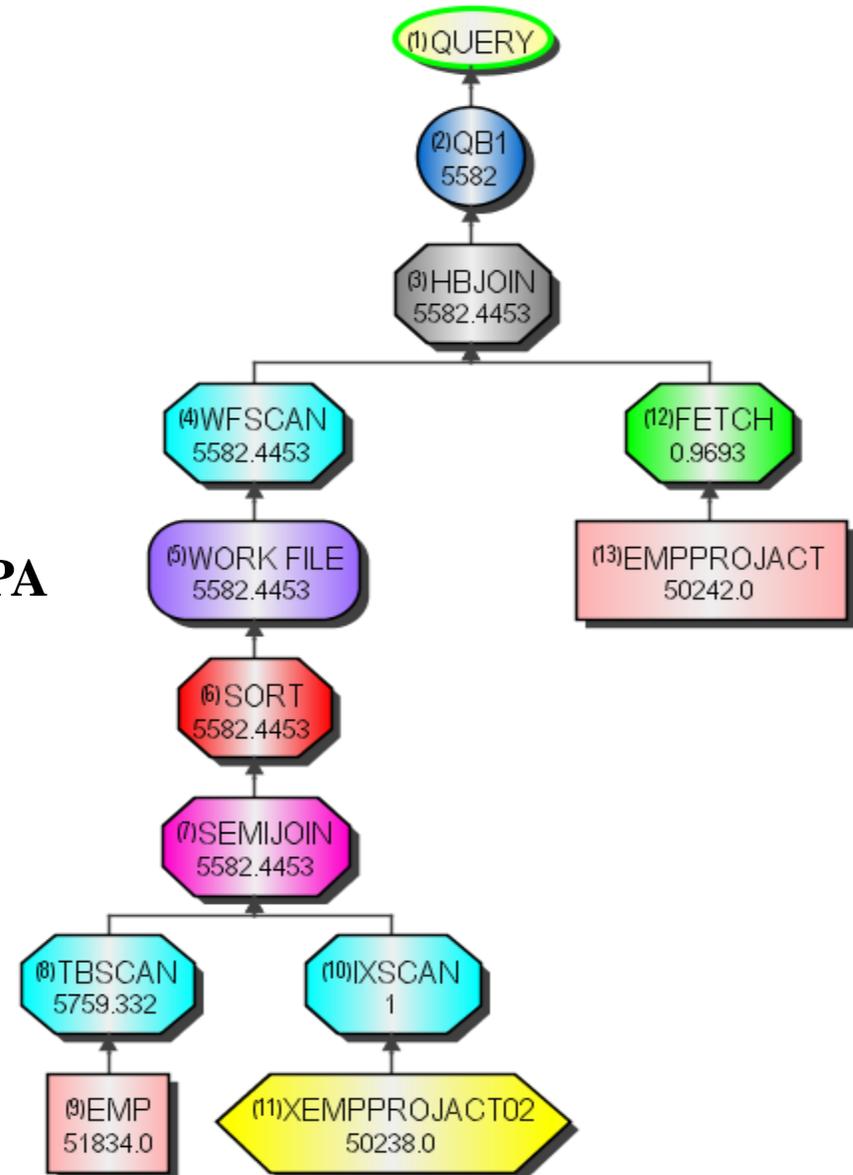


z/OS Merge Scan Join



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**

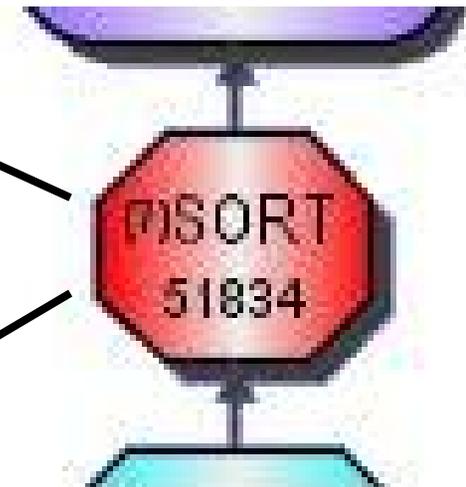
RID Sorts

- ✓ **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

Overview of Diagram
Description of Selected Node
Displays information about the node that is highlighted in the diagram.

- sort
 - Stream
 - Sort Columns
 - sortkey

Attributes

NAME	VALUE
Operator Identifier	3
Operator Type	SORT
Estimated Output Cardinality	51834
Cumulative Total Cost	5363.65
Cumulative CPU Cost	2.26175e+008
Cumulative IO cost	1860
Cumulative Re-execution Total Cost	0
Cumulative Re-execution CPU Cost	0
Cumulative Re-execution IO cost	529

Query Plan Nodes:

- (1)RETURN 5913.68
- (2)TBSCAN 5913.68
- (3)SORT 5363.65
- (4)TBSCAN 1346.18
- (00)EMP THEMIS81

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' ;
```

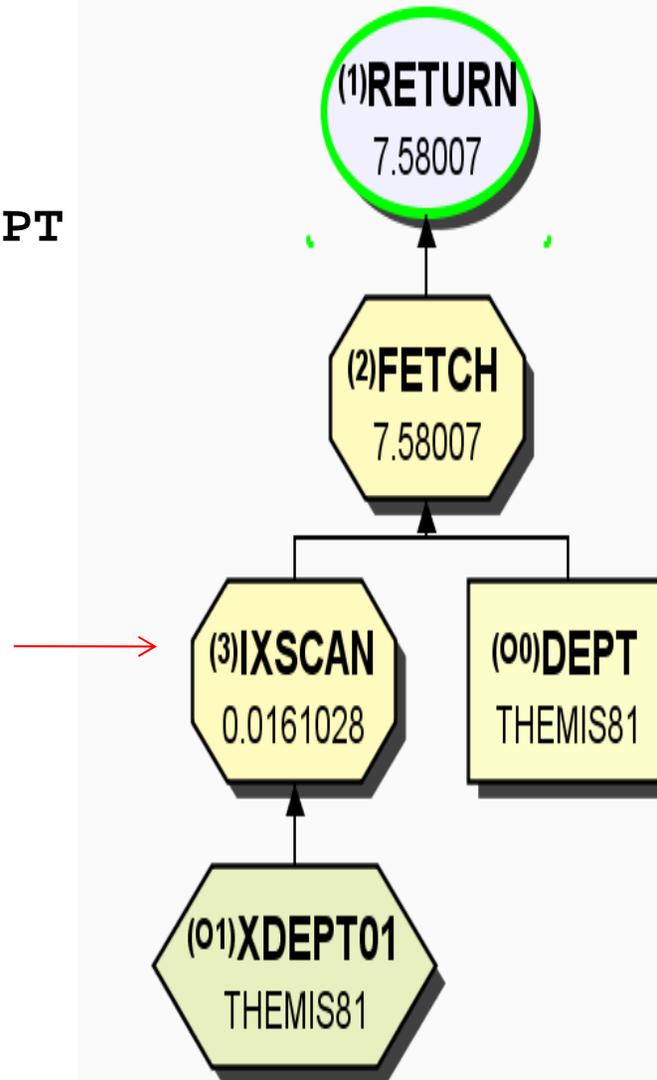
XDEPT1 index chosen !



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

The screenshot displays the DB2 Performance Center interface. On the left, the 'Overview of Diagram' pane shows a tree view with a 'return' node selected. Below it, the 'Attributes' pane lists various performance metrics for the selected node.

NAME	VALUE
Operator Identifier	1
Operator Type	RETURN
Estimated Output Cardinality	0.25
Cumulative Total Cost	7.58007
Cumulative CPU Cost	77249
Cumulative IO cost	1
Cumulative Re-execution Total Cost	0.00747882

The 'SQL Statement' dialog box is open, showing the following SQL query:

```
SELECT *
FROM DEPT
WHERE DEPTNO = ADMRDEPT
AND ADMRDEPT = 'A00'
```

The 'Optimized' radio button is selected in the dialog box. The diagram on the right illustrates the query plan structure:

- (1)RETURN 7.58007**: The final operation, highlighted with a green circle.
- (2)FETCH 7.58007**: The operation immediately preceding the return.
- (3)IXSCAN 0.0161028**: An index scan operation that feeds into the fetch.
- (00)DEPT THEMIS81**: The base table being scanned.
- (01)XDEPT01 THEMIS81**: An auxiliary table or index structure that feeds into the index scan.



Tuning a Query

Connection: DA1B

Configuration | Validation | Special Registers

Connection: DA1B Select..

Run method: JDBC

Run options:

- Refresh explorer view after script is run
- Open new connection when script is run

Tune a Query

```
Select *  
From emp  
Where Lastname = 'Smith'  
and Firstname = 'Joe'  
and Deptno = 'A00'
```



Tuning a Query

The screenshot shows the 'Query Tuner Workflow Assistant' interface. The sidebar on the left has five steps: 1. Status, 2. Capture, 3. Manage, 4. Invoke, and 5. Review. Under '2. Capture', 'Run Advisors and Analysis Tools' is selected. The main panel is titled 'Run Single-Query Advisors And Analysis Tools' and contains the following configuration options:

- Database connection: DA1B (DB2 for z/OS V10 (New-Function Mode))
- SQLID: ODYTA
- Schema: THEMIS81
- Use upper case for the SQLID and schema
- Re-EXPLAIN the query
- EXPLAIN options and runtime environment options (expanded)
- Select What To Run... Run SQL

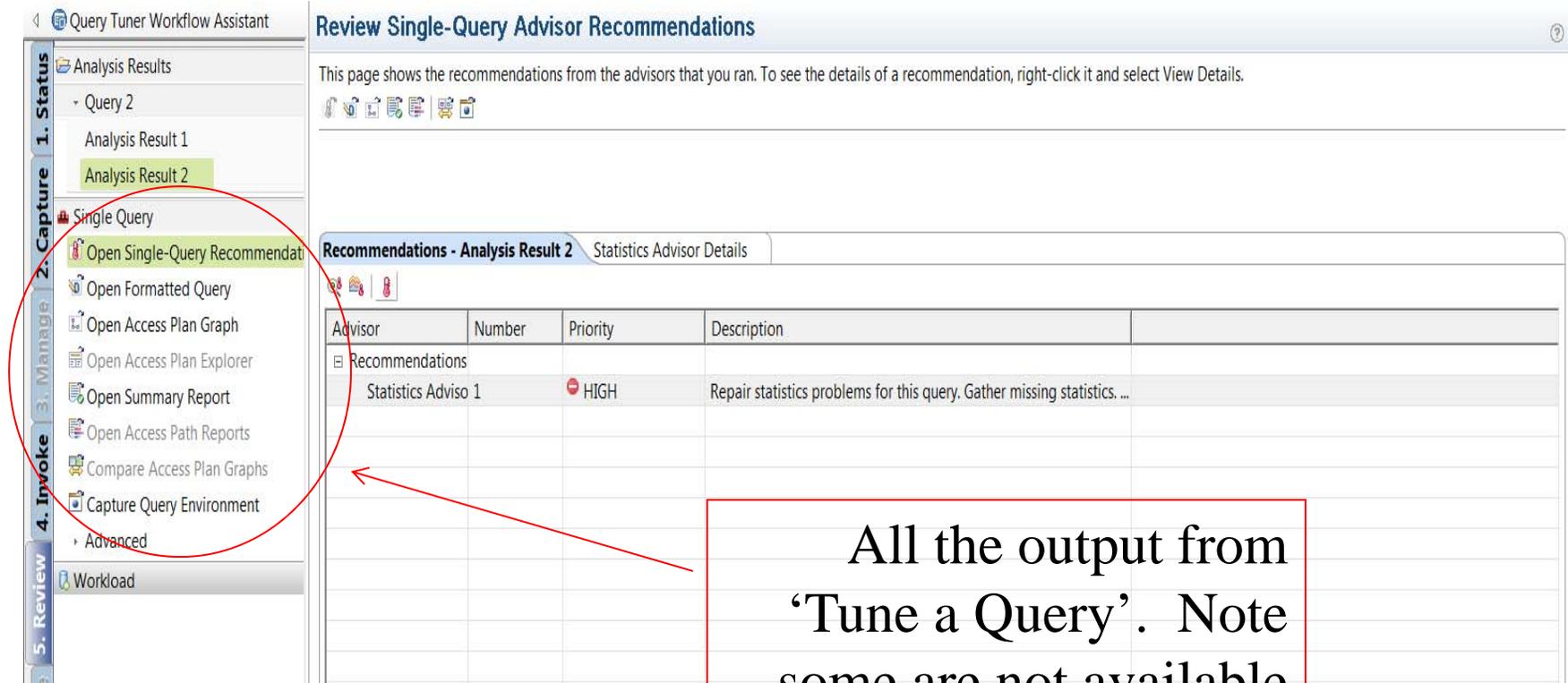
The query text area at the bottom contains the following SQL:

```
Select *
From emp
Where Lastname = 'Smith'
and Firstname = 'Joe'
and Deptno = 'A00'
```

Red circles and arrows highlight the 'SQLID' and 'Schema' fields, and the 'EXPLAIN options and runtime environment options' section.



Tuning a Query Output



Query Tuner Workflow Assistant

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

Advisor	Number	Priority	Description
Recommendations			
Statistics Adviso 1		HIGH	Repair statistics problems for this query. Gather missing statistics. ...

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



Statistics Advisor

Query Tuner Workflow Assistant

1. Status
Analysis Results
- Query 2
Analysis Result 1
Analysis Result 2

2. Capture
Single Query
Open Single-Query Recommendation
Open Formatted Query

3. Manage
Open Access Plan Graph
Open Access Plan Explorer
Open Summary Report
Open Access Path Reports

4. Invoke
Compare Access Plan Graphs
Capture Query Environment
Advanced

5. Review
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

Advisor	Number	Priority	Description
Recommendations			
Statistics Adviso 1		HIGH	Repair statistics problems for this query. Gather missing statistics. ...

Opens Advisor details tab to view information

Click on this line to expand statistics recommendations

Report

Query Tuner Workflow Assistant

1. Status

- Analysis Results
 - Query 2
 - Analysis Result 1
 - Analysis Result 2

2. Capture

- Single Query
 - Open Single-Query Recommendation
 - 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

3. Manage

- Workload

4. Invoke

- Workload

5. Review

- Workload

6. Compare

- 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. Collect 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("FIRSTNAME")
          COLGROUP("FIRSTNAME") 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

Query Tuner Workflow Assistant

Review Formatted Query

To help you understand how the query is processed, the formatted query shows a se
To the right of the query, you can view statistics for the object referenced in each se

Original Transformed

The query that was captured is shown below.

Annotations to display: All

Formatted Query	Annotation
SELECT *	
FROM THEMIS82.EMP	
WHERE (THEMIS82.EMP.DEPTNO = 'A00'	
AND THEMIS82.EMP.LASTNAME = 'Smith'	
AND THEMIS82.EMP.FIRSTNME = 'Joe'	
)	



z/OSTune a query – Query Transformation

Original Transformed

Annotations to display: All

Formatted Query

```
SELECT THEMIS81.PROJ.PROJNO
      , THEMIS81.PROJ.PROJNAME
FROM THEMIS81.PROJ
WHERE THEMIS81.PROJ.PROJNO IN (
      SELECT THEMIS81.EMPPROJECT.PROJNO
      FROM THEMIS81.EMPPROJECT
      WHERE THEMIS81.EMPPROJECT.EMENDATE > '01/01/1983'
)
```

**Note:
Non Correlated**

Original Transformed

Annotations to display: All

Formatted Query

```
SELECT THEMIS81.PROJ.PROJNO
      , THEMIS81.PROJ.PROJNAME
FROM THEMIS81.PROJ
WHERE THEMIS81.PROJ.PROJNO IN (
      SELECT THEMIS81.EMPPROJECT.PROJNO
      FROM THEMIS81.EMPPROJECT
      WHERE ( THEMIS81.EMPPROJECT.EMENDATE > '1983-01-01'
            AND THEMIS81.PROJ.PROJNO = THEMIS81.EMPPROJECT.PROJNO
      )
)
```

**Note:
Correlated**



Saving off an Access Path

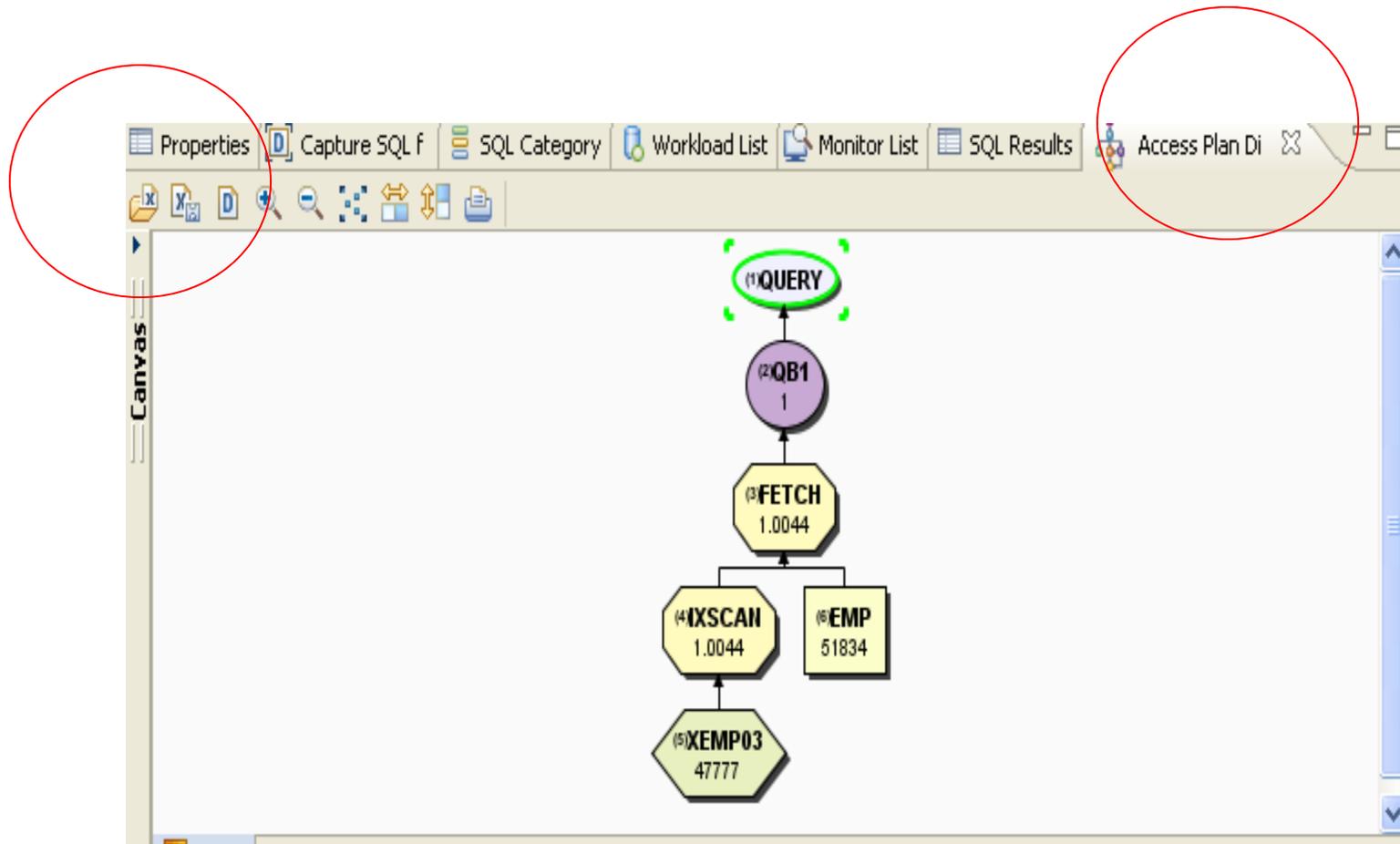
The screenshot shows the 'Access Plan Graph' window. A red circle highlights the 'Canvas' toolbar, where a context menu is open with the option 'Save the graph to a file'. The main area displays an access plan graph with the following nodes:

- (1) QUERY (green oval)
- (2) QB1 1 (purple circle)
- (3) FETCH 1.0044 (yellow hexagon)
- (4) XSCAN 1.0044 (yellow hexagon)
- (5) XEMP03 4777 (yellow hexagon)
- (6) EMP 51834 (yellow rectangle)

The graph structure is as follows: (1) QUERY is connected to (2) QB1 1. (2) QB1 1 is connected to (3) FETCH 1.0044. (3) FETCH 1.0044 is connected to (4) XSCAN 1.0044 and (6) EMP 51834. (4) XSCAN 1.0044 is connected to (5) XEMP03 4777.



Opening an Access Path



Retrieving Queries From Packages

Query Tuner Workflow Assistant

1. Status

- Text Sources
- Input Text
- File
- SQL Category
- Exported Workload

2. Capture

- DB2 for z/OS Sources
- Statement Cache
- Catalog Plan or Package**
- QMF
- QMF HPO
- Query Monitor

3. Manage

- User-defined SQL Repository

4. Invoke

- SQL Procedure
- Plan Table
- Statement Table
- Function Table
- View, Trigger, or SQL UDF

5. Review

- DB2 for Linux, UNIX, and Windows Sc

Prepare

Capture SQL from Catalog Plan or Package

Create or select a filter for capturing SQL statements from plans or packages on the subsystem. Advisors and Tools.

Database connection: D91A (DB2 for z/OS V9.1 (New-Function Mode))

Filter

Filter name: [Capture](#)

Create a filter for specific packages



Retrieving Queries From Packages

Query Tuner Workflow Assistant

Capture SQL from Catalog Plan or Package

Create or select a filter for capturing SQL statements from plans or packages on the subsystem. Then, click Capture to capture statements. Select a statement to analyze, tune, or both and click Invoke Advisors and Tools.

Database connection: D91A (DB2 for z/OS V9.1 (New-Function Mode))

Filter

Filter name: Capture

Captured Statements

The number of captured statements is 2. Right-click a statement and select Invoke Advisors and Tools. If workload tuning is enabled, you can create a workload from all of the statements.

STMTNO	QUERYNO	NAME	PROCMS	PROCSU	STMT_TEXT
331	331	DB2C160N	21588	33731	SELECT RTRIM (E . LASTNAME) CONCAT ' , ' CONCAT SUBSTR (E . FIRSTNME , 1 , 1) CONCAT ' : ' , D . DEPTNO , D . DEPTNAME , E . SALAR
454	454	DB2C160N	116	181	UPDATE EMP SET DEPTNO = : H WHERE EMPNO = : H

Retrieving Queries From Packages

Query Tuner Workflow Assistant

Run Single-Query Advisors And Analysis Tools

Specify EXPLAIN options and runtime environment options for the query. Then, run the enabled advisors and tools, or select which of them to run.

EXPLAIN options and runtime environment options

Query number: 331 Current degree: [dropdown]
SQLID: ODYTA Current refresh age: 0 [dropdown]
Schema: DBTHM00 Current maintained table types for optimization: [dropdown]
Optimization hint: [text area]

Use upper case for the SQLID and schema
 Re-EXPLAIN the query

Run Default Advisors and Tools Select What To Run...

Query Text - Query 1

```
SELECT RTRIM ( E . LASTNAME ) CONCAT ' , ' CONCAT SUBSTR ( E . FIRSTNME , 1 , 1 ) CONCAT ' . ' , D .  
DEPTNO , D . DEPTNAME , E . SALARY , E . BONUS INTO : H , : H , : H , : H : H , : H : H FROM EMP AS E  
INNER JOIN DEPT AS D ON E . DEPTNO = D . DEPTNO WHERE E . EMPNO = : H
```

Retrieving Queries From Statement Cache

Query Tuner Workflow Assistant

1. Status

- Non-DB2 Sources
 - Input Text
 - File
- 2. Capture**
 - SQL or Routine Editor
 - SQL Category
 - XML File
 - Optim Performance Manager Rep
- 3. Manage**
 - DB2 for z/OS Sources**
 - Statement Cache**
 - Catalog Plan or Package
 - QMF
- 4. Invoke**
 - QMF HPO
 - DB2 Query Monitor
 - User-defined SQL Repository
- Review**
 - SQL Procedure
 - Plan Table

Capture SQL from Statement Cache

Create or select a filter for capturing SQL statements from the dynamic statement cache on the subsystem. Then, click Capture Now or click Invoke Advisors and Tools.

Database connection: DA1A (DB2 for z/OS V10 (New-Function Mode))

Capture Query Environment...

Filter

Step 1: Select an existing filter or create a new filter.

Filter name:

(Optional) Step 2: Enable or disable the collection of statistics for the dynamic statement cache.

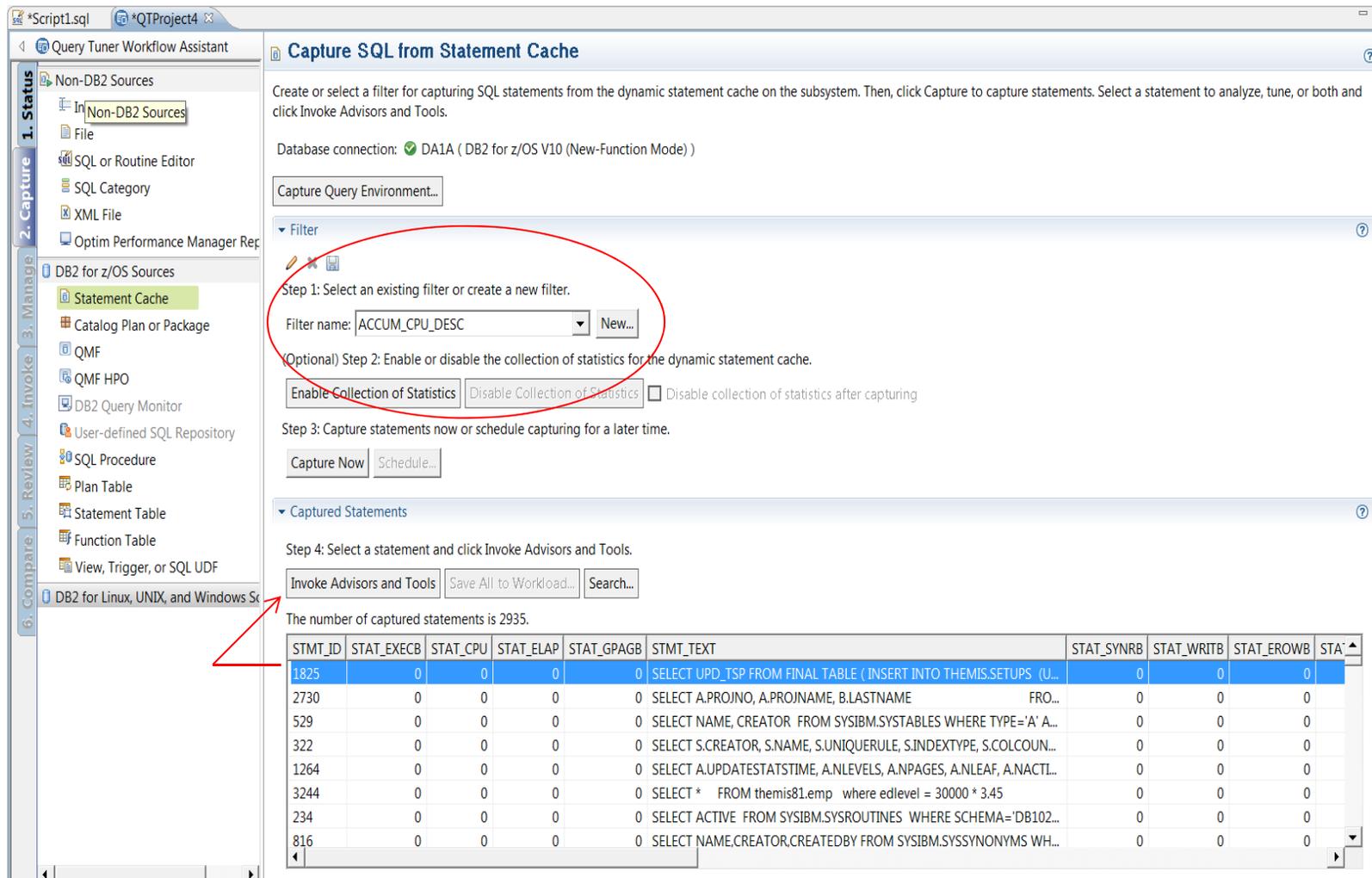
Disable collection of statistics after capturing

Step 3: Capture statements now or schedule capturing for a later time.

Create a Filter for Queries



Retrieving Queries From Statement Cache



Capture SQL from Statement Cache

Create or select a filter for capturing SQL statements from the dynamic statement cache on the subsystem. Then, click Capture to capture statements. Select a statement to analyze, tune, or both and click Invoke Advisors and Tools.

Database connection: DA1A (DB2 for z/OS V10 (New-Function Mode))

Capture Query Environment...

Filter

Step 1: Select an existing filter or create a new filter.

Filter name:

(Optional) Step 2: Enable or disable the collection of statistics for the dynamic statement cache.

Enable Collection of Statistics Disable Collection of Statistics Disable collection of statistics after capturing

Step 3: Capture statements now or schedule capturing for a later time.

Captured Statements

Step 4: Select a statement and click Invoke Advisors and Tools.

The number of captured statements is 2935.

STMT_ID	STAT_EXECB	STAT_CPU	STAT_ELAP	STAT_GPAGB	STMT_TEXT	STAT_SYNRB	STAT_WRITB	STAT_EROWB	STA'
1825	0	0	0	0	SELECT UPD_TSP FROM FINAL TABLE (INSERT INTO THEMIS.SETUPS (U...	0	0	0	
2730	0	0	0	0	SELECT A.PROJNO, A.PROJNAME, B.LASTNAME FRO...	0	0	0	
529	0	0	0	0	SELECT NAME, CREATOR FROM SYSIBM.SYSTABLES WHERE TYPE='A' A...	0	0	0	
322	0	0	0	0	SELECT S.CREATOR, S.NAME, S.UNIQUERULE, S.INDEXTYPE, S.COLCOUN...	0	0	0	
1264	0	0	0	0	SELECT A.UPDATESTATIME, A.NLEVELS, A.NPAGES, A.NLEAF, A.NACTL...	0	0	0	
3244	0	0	0	0	SELECT * FROM themis81.emp where edlevel = 30000 * 3.45	0	0	0	
234	0	0	0	0	SELECT ACTIVE FROM SYSIBM.SYSROUTINES WHERE SCHEMA='DB102...	0	0	0	
816	0	0	0	0	SELECT NAME,CREATOR,CREATEDBY FROM SYSIBM.SYSSYNONYMS WH...	0	0	0	

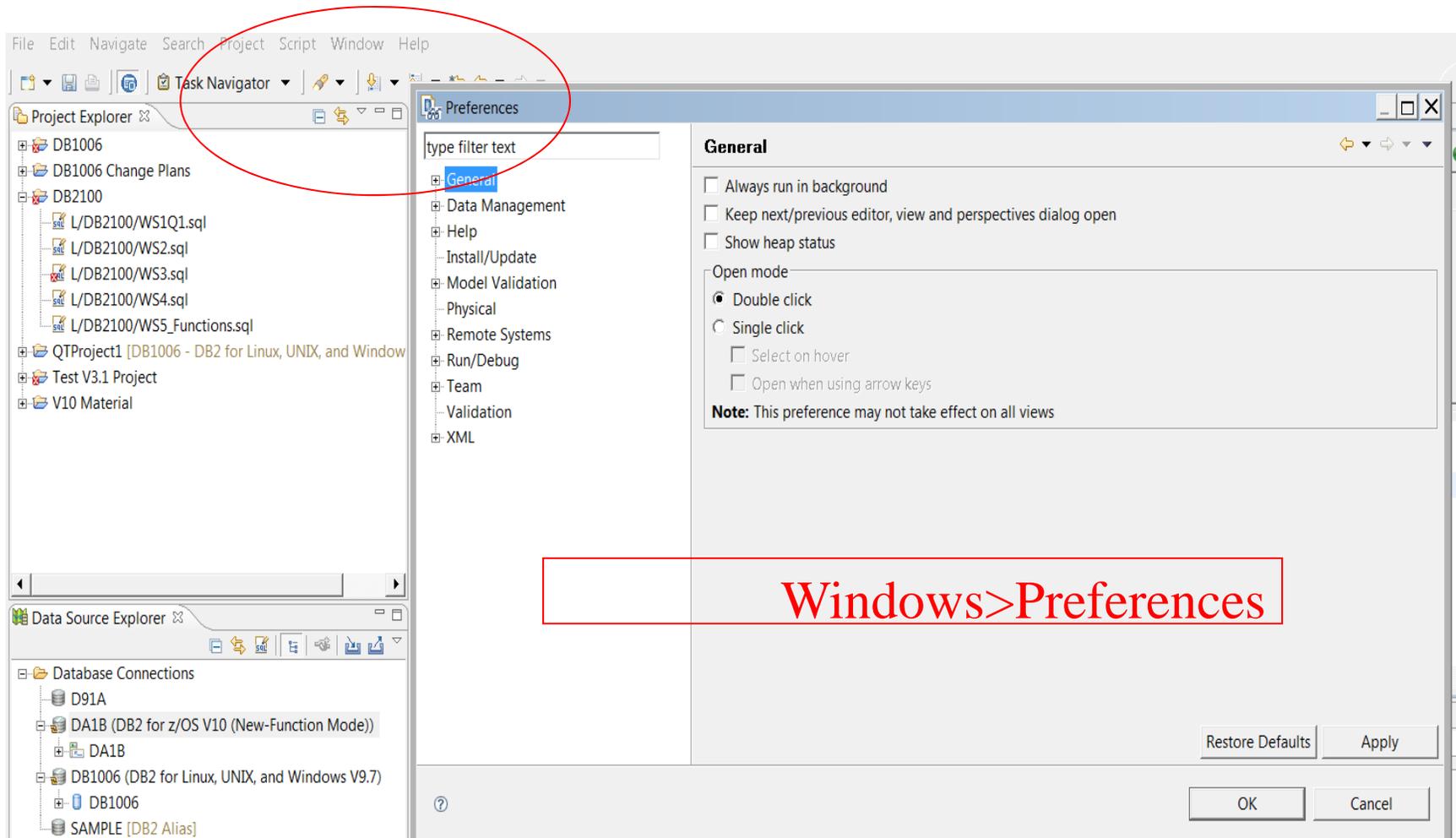
Everyone's Most Favorite Option

The screenshot displays the IBM Data Studio interface. The 'Window' menu is open, and the 'Reset Perspective...' option is highlighted with a red circle. The interface includes a Project Explorer on the left showing a project structure with folders like DB1006, DB2100, and QTProject1. The Data Source Explorer at the bottom left shows database connections including DA1B (DB2 for z/OS V10) and DB1006 (DB2 for Linux, UNIX, and Windows V9.7). The main editor window shows a SQL query:

```
Select *  
From emp  
Where Lastname = 'Smith'  
and Firstname = 'Joe'  
and Deptno = 'A00'
```

The bottom of the interface features a toolbar with icons for Database Groups, SQL Results, Access Plan Diagram, and Overview of Diagram.

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.”