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

Catalog Statistics

Object Definitions

Access Path

Rid Pools, Sort Pools

CPU Speed, # CPUs

Access Path Hint
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

‘X’ Out to Close
IBM Data Studio
Create a Project

When Creating a Project

1) Assign a Project Name and Type (Data Devlp Project
2) Assign a Subsystem
2) Assign a Default Schema
Data Studio Project Created
Data Studio Explaining Queries

Open Visual Explain
Select Subsystem Connection
Maximize Visual Explain
z/OS  Data Studio Access Path Graphs
LUW Data Studio Access Path Graphs
Data Studio Access Path Graphs

Sources of Data

(1) QUERY
(2) QB1 1
(3) FETCH 1.0044
(4) XSCAN 2
(5) XEMP03 47777
(6) EMP 51834.0
### Data Studio Access Path Graphs

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

Data Retrieval Operations
z/OS Stage 1 / 2 Predicates
Data Studio Access Path Graphs

<table>
<thead>
<tr>
<th>Input RIDs</th>
<th>51834</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Leaf Pages</td>
<td>549</td>
</tr>
<tr>
<td>Matching Predicates</td>
<td>Filter Factor</td>
</tr>
<tr>
<td>THEMIS81.EMP.LASTNAME='Smith'</td>
<td>0.0024</td>
</tr>
<tr>
<td>THEMIS81.EMP.FIRSTNAME='Joe'</td>
<td>0.0038</td>
</tr>
<tr>
<td>Scanned Leaf Pages</td>
<td>1</td>
</tr>
<tr>
<td>Output RIDs</td>
<td>2</td>
</tr>
<tr>
<td>Total Filter Factor</td>
<td>3.0149182E-5</td>
</tr>
<tr>
<td>Matching Columns</td>
<td>2</td>
</tr>
</tbody>
</table>
Data Studio Access Path Graphs

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>EMP</td>
</tr>
<tr>
<td>Creator</td>
<td>THEMIS81</td>
</tr>
<tr>
<td>Correlation Name</td>
<td>T</td>
</tr>
<tr>
<td>Type</td>
<td>T</td>
</tr>
<tr>
<td>Table Number</td>
<td>1</td>
</tr>
<tr>
<td>Qualifying Rows</td>
<td>1.0008</td>
</tr>
<tr>
<td>Base Table Type</td>
<td>T</td>
</tr>
<tr>
<td>Table Space</td>
<td>TS00EMP</td>
</tr>
</tbody>
</table>
### Tablespace Scan

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

<table>
<thead>
<tr>
<th>PLAN NO</th>
<th>METHOD</th>
<th>TNAME</th>
<th>ACCESS TYPE</th>
<th>MATCH COLS</th>
<th>ACCESS NAME</th>
<th>INDEX ONLY</th>
<th>PREFETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>EMP</td>
<td>R</td>
<td>0</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
</tbody>
</table>

**Data Studio**

Estimated number of rows

---

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

<table>
<thead>
<tr>
<th>PLAN NO</th>
<th>METHOD</th>
<th>TNAME</th>
<th>ACCESS TYPE</th>
<th>MATCH COLS</th>
<th>ACCESS NAME</th>
<th>INDEX ONLY</th>
<th>PREFETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>EMP</td>
<td>1</td>
<td>2</td>
<td>XEMP03</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

**PLAN_TABLE**

![Diagram showing database query execution plan](image_url)
z/OS Index Screening

INDEX XEMP03 on
(LASTNAME, FIRSTNAME, MIDINIT)

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

Index Screening Predicate

PLAN_TABLE

<table>
<thead>
<tr>
<th>PLAN NO</th>
<th>METHOD</th>
<th>TNAME</th>
<th>ACCESS TYPE</th>
<th>MATCH COLS</th>
<th>ACCESS NAME</th>
<th>INDEX ONLY</th>
<th>PREFETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>EMP</td>
<td>1</td>
<td>1</td>
<td>XEMP03</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
### z/OS Index Screening (cont)

**Index Scan: IXSCAN**

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

- **Show attribute explanation** selected
- **Views:** `cost_estimal`

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input RIDs</td>
<td>51834</td>
</tr>
<tr>
<td>Index Leaf Pages</td>
<td>549</td>
</tr>
<tr>
<td><strong>Matching Predicates</strong></td>
<td></td>
</tr>
<tr>
<td>THEMIS81.EMP.LASTNAME='Coldsmith'</td>
<td>0.001</td>
</tr>
<tr>
<td>Scanned Leaf Pages</td>
<td>1</td>
</tr>
<tr>
<td>Screening Predicates</td>
<td></td>
</tr>
<tr>
<td>THEMIS81.EMP.MIDINIT='R'</td>
<td>0.037</td>
</tr>
<tr>
<td>Output RIDs</td>
<td>4</td>
</tr>
<tr>
<td>Total Filter Factor</td>
<td>5.8711856E-5</td>
</tr>
<tr>
<td>Matching Columns</td>
<td>1</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>PLAN NO</th>
<th>METHOD</th>
<th>TNAME</th>
<th>ACCESS TYPE</th>
<th>MATCH COLS</th>
<th>ACCESS NAME</th>
<th>INDEX ONLY</th>
<th>PREFETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>EMP</td>
<td>1</td>
<td>0</td>
<td>XEMP03</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
z/OS Index Scan - Nonmatching

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input RIDs</td>
<td>51834</td>
</tr>
<tr>
<td>Index Leaf Pages</td>
<td>549</td>
</tr>
<tr>
<td>Scanned Leaf Pages</td>
<td>549</td>
</tr>
<tr>
<td>Screening Predicates</td>
<td>Filter Factor</td>
</tr>
<tr>
<td>THEMIS81.EMP.FIRSTNME='Michelle'</td>
<td>0.0038</td>
</tr>
<tr>
<td>THEMIS81.EMP.MIDINIT='R'</td>
<td>0.037</td>
</tr>
<tr>
<td>Output RIDs</td>
<td>9</td>
</tr>
<tr>
<td>Total Filter Factor</td>
<td>0.0002</td>
</tr>
<tr>
<td>Matching Columns</td>
<td>0</td>
</tr>
</tbody>
</table>
SELECT LASTNAME* FROM EMP
WHERE FIRSTNAME = 'David' and MIDINIT = 'A'
### Index Only Access

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

**SQL Query**

```sql
SELECT LASTNAME, FIRSTNME, MIDINIT
FROM EMP
WHERE LASTNAME LIKE 'Jo%'
```
z/OS Nested Loop Join
LUW Nested Loop Join

1. RETURN
   22.7215

2. NLJOIN
   22.7215

3. FETCH
   7.57995

4. IXSCAN
   0.0161028
   XDEPT01
   THEMIS81

5. FETCH
   22.7025

6. IXSCAN
   15.1386
   THEMIS81

0. DEPT
   THEMIS81

0.1 EMP
   THEMIS81

0.2 XEMP01
   THEMIS81
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

<table>
<thead>
<tr>
<th>Data Sorts</th>
<th>RID Sorts</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ ORDER BY</td>
<td>✓ List Prefetch</td>
</tr>
<tr>
<td>✓ GROUP BY</td>
<td>✓ Multiple Index Access</td>
</tr>
<tr>
<td>✓ DISTINCT</td>
<td>✓ Hybrid Join</td>
</tr>
<tr>
<td>✓ UNION</td>
<td></td>
</tr>
<tr>
<td>✓ Subqueries</td>
<td></td>
</tr>
<tr>
<td>✓ JOIN</td>
<td></td>
</tr>
</tbody>
</table>
**z/OS Data Sorts via Data Studio**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Cardinality</td>
<td>51834</td>
</tr>
<tr>
<td>Output Cardinality</td>
<td>51834</td>
</tr>
<tr>
<td>Pages</td>
<td>489</td>
</tr>
<tr>
<td>Record Size</td>
<td>21</td>
</tr>
<tr>
<td>Key Size</td>
<td>4</td>
</tr>
</tbody>
</table>
LUW Data Sorts via Data Studio
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

1. RETURN 7.58007
2. FETCH 7.58007
3. IXSCAN 0.0161028
   - DEPT THEMIS81
   - XDEPT01 THEMIS81
Tuning a Query

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

Run Single-Query Advisors And Analysis Tools

Specify EXPLAIN options and runtime environment options for the query. You can optionally:

- Database connection: DA1B (DB2 for z/OS V10 (New-Function Mode))
- SQUID: ODYTA
- Schema: THEMIS81
- Use uppercase case for the SQUID and schema
- Re-EXPLAIN the query

Select What To Run: Run SQL

Query Text - Query 1

```sql
SELECT *
FROM emp
WHERE Lastname = 'Smith'
    AND Firstname = 'Joe'
    AND Deptno = 'A00'
```
Tuning a Query Output

All the output from ‘Tune a Query’. Note some are not available in free version.
Click to edit Master title style

Statistics Advisor

Opens Advisor details tab to view information

Click on this line to expand statistics recommendations
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


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 this command, it can repair 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 the recommended 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

- Statistics Advisor report
- Conflicts detail
- RUNSTATS Result
z/OS Tune a query – Query Transformation

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

Original

Transformed

The query that was captured is shown below.

Annotations to display:

Formatted Query

<table>
<thead>
<tr>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SELECT</code> *</td>
</tr>
<tr>
<td>FROM THEMIS82.EMP</td>
</tr>
<tr>
<td>WHERE ( THEMIS82.EMP.DEPTNO = 'A00' AND THEMIS82.EMP.LASTNAME = 'Smith' AND THEMIS82.EMP.FIRSTNAME = 'Joe' )</td>
</tr>
</tbody>
</table>
z/OSTune a query – Query Transformation

Note: Non Correlated

Note: Correlated
Saving off an Access Path

Access Plan Graph

Canvas

Save the graph to a file
Opening an Access Path
Retrieving Queries From Packages

Create a filter for specific packages
Retrieving Queries From Packages

**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))

**Captured Statements**

<table>
<thead>
<tr>
<th>_STMTNO</th>
<th>QUERYSNO</th>
<th>NAME</th>
<th>PROCMS</th>
<th>PROCSU</th>
<th>STMT_TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>331</td>
<td>331</td>
<td>DB2C160N</td>
<td>21588</td>
<td>33731</td>
<td>SELECT RTRIM (E.LASTNAME) CONCAT ' ' CONCAT SUBSTR (E.FIRSTNAME, 1, 1) CONCAT ' ', D.DEPTNO, D.DEPTNAME, E.SALAR</td>
</tr>
<tr>
<td>454</td>
<td>454</td>
<td>DB2C160N</td>
<td>116</td>
<td>181</td>
<td>UPDATE EMP.SET DEPTNO = :H WHERE EMPNO = :H</td>
</tr>
</tbody>
</table>

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.
Retrieving Queries From Packages

Run advisors on current package or re-explain
Retrieving Queries From Statement Cache

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

Default Query Environment

Filter

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

Filter name: ACCUM_CPU_DESC

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

Enable Collection of Statistics

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

Capture Now

Captured Statements

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

The number of captured statements is 2935.
Everyone’s Most Favorite Option
Preference Settings

Windows > Preferences
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.”