

Developer Best Practices for DB2 Performance

Tony Andrews

tandrews@themisinc.com

Twitter



Follow @ThemisTraining

Questions?

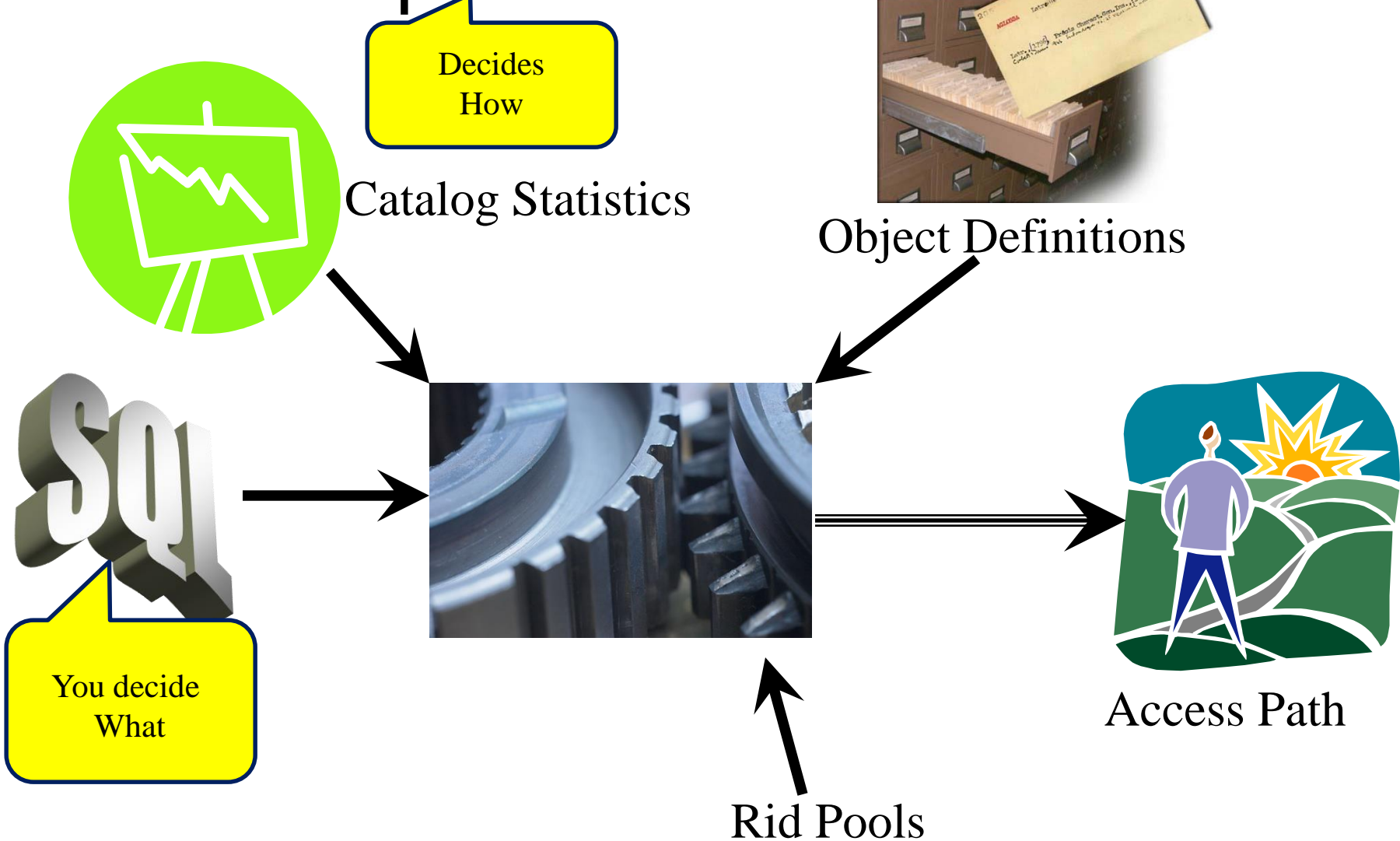
- I will try my best to get to some questions towards the end of the webinar.
- You can submit questions by typing into the questions area of your webinar control panel.
- Any questions not answered due to time constraints can be answered afterward via an email.

Webinar Objectives

- Understand the key areas that can cause performance issues with DB2 queries and applications
- Better understand DB2 optimization
- Understand how DB2 database design affects performance
- Understand DB2 data distribution statistics. How does it affect performance
- Understand SQL efficient coding

DB2 Optimization

The DB2 Optimizer



Developer Tuning Approaches

- Change Physical Design
- Gather / Alter Statistics
- Change the SQL.

***** See a prior Webinar
'The Power of the SQL Rewrite'**



DB2 Database Design

Indexes, what are they good for?

1. To enforce a primary key and/or unique constraint
2. To support of foreign keys
3. To support SQL join processing
4. To define table partitioning. An old requirement, currently an option
5. To improve performance in queries
6. **To specify the clustering order of data in table space file**

Indexes for performance

1. Design indexes that match predicates in queries
2. Read only table typically have more indexes, especially data warehouse designs.
3. The more indexes on a table, the more I/O takes place on inserts, updates, and deletes because these actions also manage index values in those index space files.
REORGS will also run longer.
4. Know your application. What are the most frequent executed queries? How often are inserts, updates, and deletes executed?
5. Queries with table space scans? How often are they run?

Indexes for clustering

A 'Clustering Index' specifies how data is physically ordered in the table space file. Operations that benefit from a good clustering design are:

- Queries that return a high number of rows
- Grouping operations
- Ordering operations
- Join operations
- Range type predicates.

EMP table clustered by EMPNO

000010 HAAS A00 000020 THOMPSON B01 000030 KWAN C01 000050 GEYER E01 000060 STERN D11 000070 PULASKI D21 000090 HENDERSON E11	000100 SPENSER E21 000110 LUCHESI A00 000120 O'CONNELL A00 000130 QUINTANA C01 000140 NICHOLLS C01 000150 ADAMSON D11 000160 PIANKA D11	

Should this table be in EMPNO Primary Key order?

It Depends.....

EMP table clustered by EMPNO

000010 HAAS A00 000020 THOMPSON B01 000030 KWAN C01 000050 GEYER E01 000060 STERN D11 000070 PULASKI D21 000090 HENDERSON E11	000100 SPENSER E21 000110 LUCHESI A00 000120 O'CONNELL A00 000130 QUINTANA C01 000140 NICHOLLS C01 000150 ADAMSON D11 000160 PIANKA D11	

What happens here?

```
SELECT *  
FROM EMP  
WHERE DEPTNO = 'A00'
```

Where are all the rows that
have 'A00' as a DEPTNO value?

IF there were 100 rows that contain
this value, they could be on 100
pages of data. Yes?

DB2 Data Distribution Statistics

DB2 Runstats Utility

'Runstats' is a utility that will put data distribution statistics in the DB2 SYSIBM catalog tables for a specific table.

- Optimizer takes into account table statistics for access path optimization
- Cardinality statistics should be generated for each column
- Cardinality of total rows are generated for the table
- Frequency value statistics are great for data with very uneven distribution of values
- Histogram statistics great for range type predicates
- Hard coding and/or dynamic queries needed to take advantage of Frequency / Histogram statistics

DB2 Runstats Utility – Freqval Stats

Example: COMM column in EMP table has 31 values, but the value 0 is 99% of the data.

Frequency value statistics loads this information for DB2

```
SELECT *  
FROM EMP  
WHERE LASTNAME LIKE 'S%'  
AND COMM = 0
```

```
SELECT *  
FROM EMP  
WHERE LASTNAME LIKE 'S%'  
AND COMM = ?
```

```
SELECT *  
FROM EMP  
WHERE LASTNAME LIKE 'S%'  
AND COMM = ?  
AND COMM <> 0
```

Coding Efficient SQL

Writing efficient SQL queries

1. Are SQL statements coded as simply as possible?
2. Are filtering predicates coded efficiently?
3. Are there any Stage 2 predicates? Can they be recoded
4. a different way?
5. Are there any non-indexable predicates? Can they be recoded a different way?
6. Are Joins coded efficiently?
7. Are Subqueries coded appropriately?
8. Are there any aggregate functions?

SQL Performance Basics

- Know your Data
- Understand how the data is used
- Write efficient SQL Queries
- RUNSTATS to gather Statistics
- Validate Access Paths Chosen - EXPLAIN
- Tuning Tools:
 - ✓ Explain
 - ✓ Visual Explain via Data Studio
 - ✓ Monitors & Performance Traces

Thank you for allowing me and Themis
to share some of our experience and
knowledge today!

Tony Andrews

tandrews@themisinc.com

I hope that you learned something new today !!!!

The material in this presentation is further developed in the following Themis courses:

DB1032 – DB2 for z/OS Performance and Tuning

DB1041 – DB2 z/OS Advanced SQL

DB1037 – Advanced Query Tuning using IBM
Data Studio

DB1051 – High Performance Application Design

DB1006 – DB2 LUW Advanced Query Tuning using
IBM Data Studio

Links to these courses may be found at: www.themisinc.com

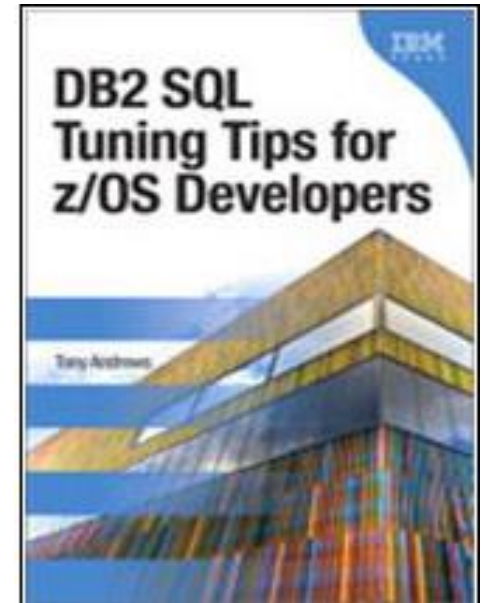
Tony's Email: tandrews@themisinc.com

Twitter: [@ThemisTraining](https://twitter.com/ThemisTraining)

Education. Check out
www.db2sqltuningtips.com
www.ibmpress.com
www.amazon.com

Finally! A book of DB2 SQL tuning tips for developers, specifically designed to improve performance.

DB2 SQL developers now have a handy reference guide with tuning tips to improve performance in queries, programs and applications.



Education. Check Out www.themisinc.com

- On-site and Public
- Instructor -led
- Hands-on
- Customization
- Experience
- Over 30 DB2 courses
- Over 400 IT courses



US 1-800-756-3000

Intl. 1-908-233-8900

