

Transactional vs Analytical SQL

Rob Bestgen
(bestgen@us.ibm.com)
IBM - Db2 for i Consultant

© 2018 IBM Corporation

IBM Power Systems



Modern Database Defined

- A modern database contains data that is accurate and complete
- The data is organized to be retrieved when needed and in the format required
- The design of the database is well planned
- The database is tailored to the business needs

Reference:

Jan L Harrington (1998) *Relational Database Design Clearly Explained*

Getting Information

Accessing data for information can be done in a myriad of ways.

But it generally comes down to two main categories:

- Specific questions for targeted information, usually paired with subsequent modifications (inserts/updates)
- General questions looking at things from a higher level perspective

OLTP and OLAP

Specific, targeted questions are associated with ***OnLine Transactional Processing*** or *OLTP*

General questions usually fall under the category of ***OnLine Analytical Processing*** or *OLAP*






The two can sometimes be mixed together in a single application

- When mixed, it is often incorporating OLAP to provide supplemental information in an OLTP environment

OLAP environments are frequently called *reporting environments*

Two Worlds, frequently colliding

Classify the following query requests as either OLTP or OLAP

- Lookup order information for Ms. Jones for last Tuesday 
- How many total orders have been taken year-to-date, by month, classified by major product category and state in which they were sold? 
- Select and print out all orders for today 
- Compare revenue from the 1st quarter of this year to last year 
- Show a summary of a customer's account activity in the last year 

Noting the differences

OLTP

- Small, focused requests involving one or a small number of rows
- Specific, detailed information being requested
- Can also include repetitive operations e.g. update all orders of widgets as now being shipped

OLAP

- Asking questions at a higher level, not individual entries
- Often involves many rows of data to answer the question (query)
- Almost always aggregates data (GROUP BY, DISTINCT...)

Noting the differences

OLTP database layout

- (Potentially) highly normalized
 - ‘chained’ e.g. order header, order detail
 - ex: retail customers in one file, wholesale in another
- Based around application needs
- Potentially embedded meanings in a field
 - If flg = 1 then cnt is num units, else cnt is num cases
- In some cases just glorified, persistent arrays
- ACID properties required

OLAP database layout (preferred)

- Organized along dimensions
 - customers/locations
 - product hierarchy
 - date hierarchy
 - ...
- Facts (things that are counted) are organized and ‘easy’ to summarize
- Getting individual information is possible but not necessarily efficient
- Transaction boundaries can be fuzzy

Why does it matter?

Does it really matter whether its OLTP or OLAP?

- Database requirements can be dramatically different
 - Indexing, memory, CPU
- OLAP can consume copious resources and disrupt other work
- OLAP over the wrong database design can make performance tuning difficult, even with indexing
- Size matters
 - for small tables (less than 1M rows), OLAP performance is often not a concern

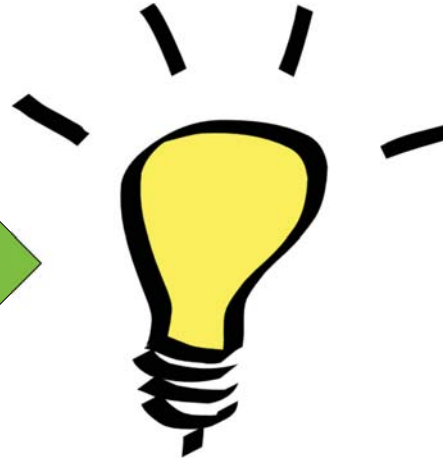
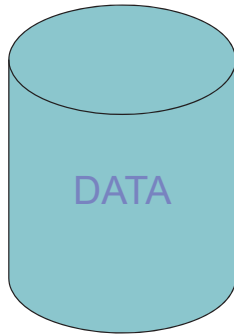
Warning Signs

Some potential warning signs (in your SQL queries) that your database file design may not be right for your purposes

- Common use of derivations (case, substr, concat...)
 - *where date('20' concat dfld) > '2018-05-01'*
- Use of **ORs** in WHERE or ON clause that involve different fields
 - *where status = 'A' or code = 999*
- Heavy use of operators other than = or **IN** in WHERE/ON clause
 - Date fields excluded
- Common use of (QTEMP) work files to contain intermediate work
- Frequent use of UNION/UNION ALL
- Seemingly impervious to performance tuning

Different Approaches

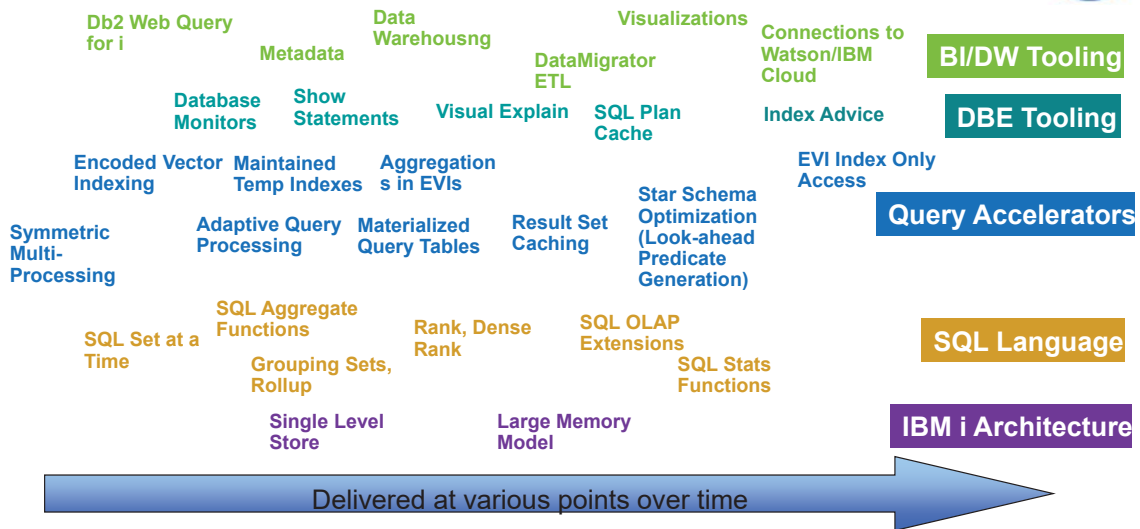
Turning Data into Insight



OLAP is Online *Analytical* Processing

Rich function available directly on Db2 for i

IBM i Business Intelligence/Analytics Supporting Attributes



Analytics solution

Db2 Web Query for i!

- From Report Modernization to Business Intelligence with Advanced Visualizations
- Operational Reporting or Automate Building of Data Warehouse/ Data Mart Infrastructures
- Simplify Report Authoring by untangling data and documenting relationships/rules
- Deliver information to those that need it the most, in the forms they desire – while maintaining “single version of the truth”
 - Dashboards
 - Spreadsheets
 - PDF
 - Mobile Devices
 - Embedded in your own applications



Legacy Date

Orddate – P8 Decimal
02132016
03212017
06012016

Date Usage Desired by End Users

- Order Date Month
- Order Date Year
- Order Date Quarter
- Order Date Day Name
- Current Date
- Current Month
- Current Year
- Today's Date – 30
- Same Date LastYear
- Is it a Weekend?
- Is it a Holiday?
- What was the weather?
- Was there a full moon?



Db2 Web Query Installation and Setup (in 15 minutes)

With the NEW **EZ-Install Package**



Request by sending email to QU2@us.ibm.com w/ serial number of system where you plan to install

Approaches

Different solutions

- Tune
- Minimize/eliminate virtual transformations
- Leverage power of SQL and tools
- Isolate
- Move and transform

Tuning

Often some basic tuning can make a huge difference. Use SQL performance tuning best practices

- Plan Cache analysis
- Index advised
- Visual Explain

Tuning

SQE Plan Cache

The screenshot shows the IBM System i Navigator interface. On the left is the 'System i Navigator' tree view showing a hierarchy of connections and databases. The main window displays the 'Properties' for the 'SQE Plan Cache'. A context menu is open over the 'SQE Plan Cache' entry, with 'Show Statements...' highlighted. A blue callout bubble with the text 'Show Queries' points to this menu item.

Description	Value	Value Unit
Time Of Summary	2018-04-18-05.58.40.67691	
Plan Cache Creation Time	2018-04-11-01.27.18.965488	
Active Query Summary		
Number of Currently Active Queries	342	
Number of Queries Run Since Start	140246	
Number of Query Full Opens Since Start	80285	
Plan Usage Summary		
Current Number of Plans in Cache	1692	
Total Number of Plans Built Since Start	2722	
Total Number of Unique Queries Since Start	1580	
Current Plan Cache Size	168	MB
Current Plan Cache Size Threshold	*AUTO	
Maximum Plan Cache Size For Autosizing	*DEFAULT (6144)	MB
Current Plan Cache Hit Ratio	96	%
Target Plan Cache AutoSize Ratio	*DEFAULT (90)	%
Current Number of Job Scoped (EMP) Plans	32	
Total Number of Job Scoped (EMP) Plans	38	
Total Number of Unique Queries with Job Sc...	11	
Total Times Plans Used from Cache	77564	
Total Plans Pruned	0	
Current Number of Temporary Objects	1491	
Current Total Size of Temporary Objects	352	MB
Maximum Number of Temporary Objects	*DEFAULT (5)	
Total Number of Temporary Objects Created	24	
Current Number of Temporary Objects in Cache		
Number of Plans Rebuilt		
Plan Cache Configuration		
Current Plan Cache Size	168	MB
Maximum Plan Cache Size	6144	MB
Target Plan Cache Size	90	%
Maximum Number of Temporary Objects	3	
Done: 38 rows		

17

© 2018 IBM Corporation

Handling expressions

- Pay attention to datatypes on mapping or comparison
 - Host variable attribute matching, joining on matching data type columns....
- Let the database do the CASTing instead of you
 - Don't add a CAST for comparison unless you need to
- Cast the non-column instead of the column in a comparison, if a cast is needed

Ex:

```
where tmspfld > timestamp(current date)
```

is better than

```
Where date(tmspfld) = current date
```
- Can create a derived key index when the expression is necessary

Ex:

```
create index xxxx on mytable(date(tmspfld))
```

Note: derived key indexes are not advised!

18

© 2018 IBM Corporation

Clean up using lookup tables

What Are Lookup Tables?

- Tables that are joined to enhance existing data
 - Can be used to store output of pre-calculated functions
 - Date attributes (day of week, fiscal year, etc)
 - Most often used to translate encodings into user-friendly descriptions
 - Date formats
 - Country codes
 - Eliminates performance overhead of dynamically computing different descriptions and calculations
 - Also known as Dimension, Auxiliary or Associative tables
- Typically joined using an existing foreign key relationship
 - Foreign key not required
 - SQL View can be used to mask join complexity or facilitate reuse

Enhancing legacy Date fields with Lookup table

Date Lookup Table methodology:

1. Create & Populate the date lookup table
(Example SQL statement available)
2. Add date lookup table join to applications & reports
3. Validate right indexes are in place for the join



Enhancing legacy Date fields with Lookup table

Legacy ORDERS file →

ORDER	CUST	ORDDAT	SHPDAT	SHPVIA	ORDSTS	ORDAMT	TOTLIN
1715810	H4541	2152005	6302005	UPS	1	933346.39	2
1563685	R1948	2202005	8042005	Mule Train	2	0.00	0
7195900	Q7881	2232005	5022005	Pick Up	2	0.00	0
8854635	S1511	2232005	9102005	Train	1	118086.88	2
6694902	X8863	2242005	7192005	Camel	2	0.00	0
8054679	F4327	2272005	9112005	Pnuematic Tube	2	0.00	0
527879	C3233	2282005	5022005	US Mail	1	897524.71	3
4011038	G1496	2282005	10162005	Train	1	865114.60	2
5417918	J3825	2282005	10022005	FedEx	1	84608.50	2
9456994	N2796	2282005	7012005	Train	2	0.00	0
3526155	Q7881	3012005	8062005	UPS	1	194660.67	2

Date lookup table

JOIN

DC_DATE	DC_MDYY_DEC	DC_YEAR	DC_DOW	DC_DOY	DC_MM	DC_DD	DC_DAY_NAME	DC_WEEKEND	DC_FISCAL_YEAR	DC_FISCAL_QUARTER	DC_MONTH_NAME	DC_MONTH_ABRV	DC_SEASON
2005-02-18	2182005	2005	5	49	2	18	Friday	N	2005	2	February	FEB	Winter
2005-02-19	2192005	2005	6	50	2	19	Saturday	N	2005	2	February	FEB	Winter
2005-02-20	2202005	2005	7	51	2	20	Sunday	Y	2005	2	February	FEB	Winter
2005-02-21	2212005	2005	1	52	2	21	Monday	Y	2005	2	February	FEB	Winter
2005-02-22	2222005	2005	2	53	2	22	Tuesday	N	2005	2	February	FEB	Winter
2005-02-23	2232005	2005	3	54	2	23	Wednesday	N	2005	2	February	FEB	Winter
2005-02-24	2242005	2005	4	55	2	24	Thursday	N	2005	2	February	FEB	Winter
2005-02-25	2252005	2005	5	56	2	25	Friday	N	2005	2	February	FEB	Winter
2005-02-26	2262005	2005	6	57	2	26	Saturday	N	2005	2	February	FEB	Winter
2005-02-27	2272005	2005	7	58	2	27	Sunday	Y	2005	2	February	FEB	Winter
2005-02-28	2282005	2005	1	59	2	28	Monday	Y	2005	2	February	FEB	Winter
2005-03-01	3012005	2005	2	60	3	1	Tuesday	N	2005	2	March	MAR	Winter
2005-03-02	3022005	2005	3	61	3	2	Wednesday	N	2005	2	March	MAR	Winter
2005-03-03	3032005	2005	4	62	3	3	Thursday	N	2005	2	March	MAR	Winter

Example Of Using Date Lookup table with View

```
CREATE VIEW Facts_and_Dates AS
SELECT F.*, C.*
FROM Facts F INNER JOIN Calendar_master C
ON Start_date = Numeric_Date
```

Calendar_Master	
Column	Attributes
Date_ID	PRIMARY KEY
Date_col	DATE
Numeric_Date	NUMERIC(8,0)
ISO_Date	CHAR(8)
Day_of_Week	CHAR(3)

Facts	
Column Name	Column type
ID	INTEGER
Status	CHAR(1)
Start_Date	NUMERIC(8,0)
More...	

```
SELECT * FROM Facts_and_Dates
WHERE Day_of_Week = 'MON'
```

Leverage the Power of SQL

Are you making the best use of SQL?

- Set at a time processing?
- Limiting duplication and complexity with Common Table Expressions (CTEs)?
- Using advanced capabilities like GROUPING SETS and OLAP functions to solve problems with simpler, more powerful solutions?
- Having the database do more for you?

SQL Reference:

http://ibm.biz/DB2fori_SQLreference

```

1 |-- Generate SQL
2 -- Version:                VTRM0 160422
3 -- Generated on:           01/19/17 09:49:56
4 -- Relational Database:    DB/COE1
5 -- Standards Options:     DB2 For i
6 CREATE VIEW SAMPLES.CHPT0AVG (
7   EMPNO,
8   WORKDEPT,
9   SALARY,
10  ABOVE_OR_BELOW_AVG,
11  DEPT_AVG_SALARY,
12  )
13 AS
14
15 SELECT EMPNO, WORKDEPT, SALARY,
16        CASE
17          WHEN (SALARY - (AVG(SALARY) OVER (PARTITION BY WORKDEPT))) > 0
18             THEN '...IS ABOVE AVG'
19          WHEN (SALARY - (AVG(SALARY) OVER (PARTITION BY WORKDEPT))) < 0
20             THEN '...IS BELOW AVG'
21          ELSE '...IS EQUAL TO AVG'
22        END AS ABOVE_OR_BELOW_AVG,
23        DECIMAL(AVG(SALARY) OVER (PARTITION BY WORKDEPT)) AS DEPT_AVG_SALARY,
24        DECIMAL(SALARY - (AVG(SALARY) OVER (PARTITION BY WORKDEPT))) AS DELTA
25 FROM   SAMPLES.EMPLOYEE;

```

Manage

Isolate/balance

- Schedule work for off hours
- Put reports in their own subsystem
 - potentially leverage workload capping
- Separate memory pools
- Balance resource allocation according to business priority
- Put data and reporting on its own system
 - Operational Data Store (ODS)

Transform

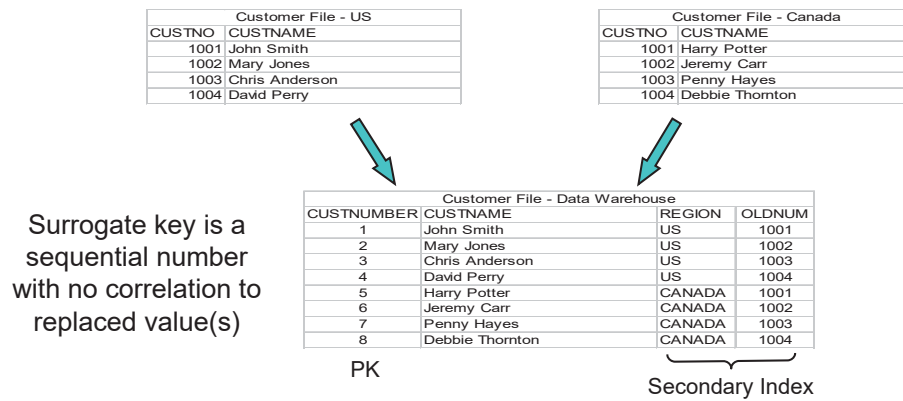
Performance

- Many BI (OLAP) queries are complex and **summary in nature**
 - “Show me revenue vs. forecast across these 3 product lines, these 2 regions, and the trend over the last 3 quarters.”

- These kinds of queries, against OLTP designed databases, on systems tuned for transaction processing, **can be problematic!**
 - Data not in multidimensional formats
 - Data in detail level, not summary
 - Data transformation work is being done at report run time
 - Data consolidation done at run time
 - Data is point in time, not historical (usually)



Difficult Data Issues



Difficult Data Issues

- Multiple application databases
 - May even be different types (Db2, Oracle, SQL Server)
 - Data stored in totally different structures – but related information



- Very difficult, if not impossible to join tables across databases
- Different security, availability etc
- Adds significant complexity

Consider

If you have to manipulate data to get it into the proper form, you are **transforming** it

- Every time it is accessed
- Which means the same steps are repeated for each user
- And what if the user encounters data that isn't valid?

So consider

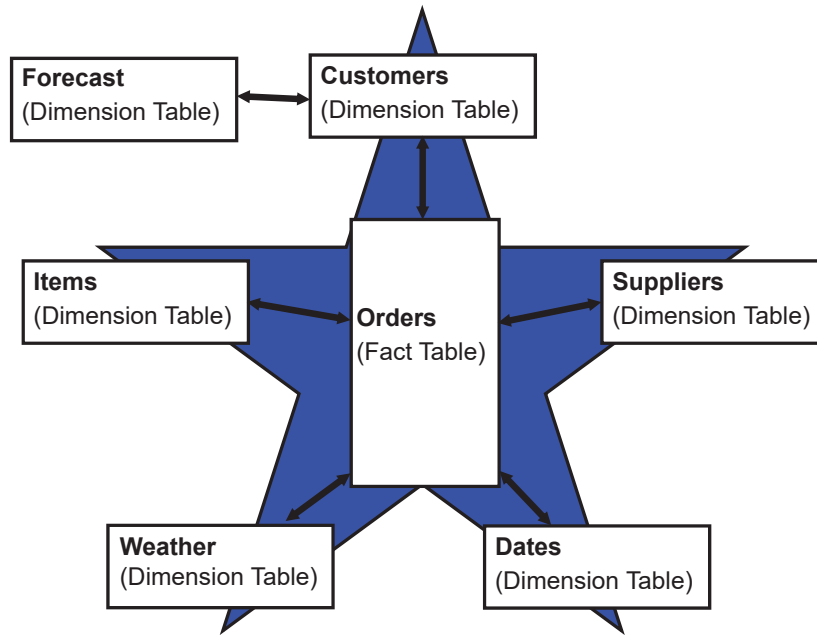
- Transform it once and let everyone use that transformed version
- Performance improves? Well, yeah!
- Verify data is clean and prep'd for use

Answer: The Data Warehouse

- Automate Data Consolidation
- Move Data Transformations to the ETL process
 - Automate this process to run in the background
 - Validate data as it enters the warehouse
- Transform the data into an analytical data model
 - E.g., Star Schema
- Automate Capturing and Maintaining Historical Data
 - Slowly changing dimensions
- Optimize Environment for Reporting
 - E.g., Create Summaries



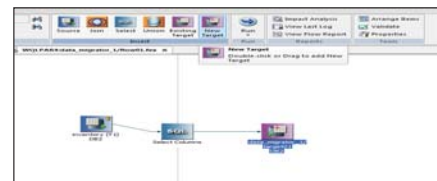
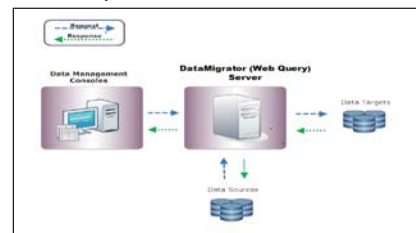
BI/OLAP - Star Schema Data Model



ETL Tool Example

DataMigrator ETL (Extract, Transformation, Load)

- A product associated with Db2 Web Query
- Product ID 5733-WQM
 - Requires either Db2 Web Query Express or Standard Edition
 - Core based (must have same number of cores as DB2 Web Query)
 - Standard Edition required for accessing non DB2 relational databases as a data source
 - One license of DB2 Web Query Developer Workbench is required to set up data and process flows
- ETL for IBM i
 - Like DB2 Web Query, all server code and ETL objects run in and/or are stored in IBM i
- INTEGRATED with DB2 Web Query
 - Can leverage meta data created with DB2 Web Query or vice versa
 - Same look and feel
 - Shared services and administration



Send email to qu2@us.ibm.com

Next Steps

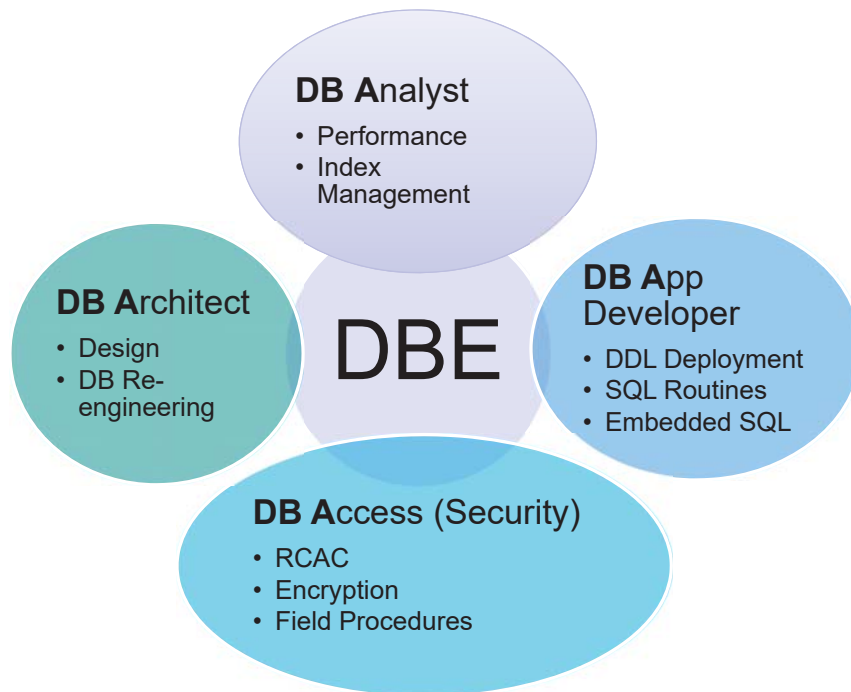
Take Action?

Should you take action to improve your OLAP workload?

- Do you have OLAP-type workloads?
- Do you recognize the warning signs in your environment?
- Have you struggled with getting good performance (and really tried with indexing)?
- Queries getting too unwieldy?
- Increasing requests for reports and analysis?

- Are you or do you have a Database Engineer (DBE)?

What is a Database Engineer (DBE)?



Summary

- There are considerable differences between OLTP and OLAP
- Knowing the difference is the first step in handling them properly
- Proper design leads to proper performance
- Data preparation is a key success factor
- Projecting a 'business view' of the data enables business users
- Consolidating data can often be needed to achieve functional and/or performance objectives for OLAP
- ETL with automation is necessary to be successful long term in consolidating data



- Are you experiencing performance problems?
- Are you using SQL?
- Are you getting the most out of Db2 for i?



Need help?

IBM Db2 for i Team

- ✓ Database modernization
- ✓ Db2 Web Query
- ✓ Database architecture and design
- ✓ Db2 SQL performance analysis and tuning
- ✓ Data warehousing and Business Intelligence
- ✓ Db2 for i Advanced SQL

Contact: Mike Cain mcain@us.ibm.com
Tom McKinley mac2@us.ibm.com
IBM Systems Lab Services
Rochester, MN USA

Thank You!