

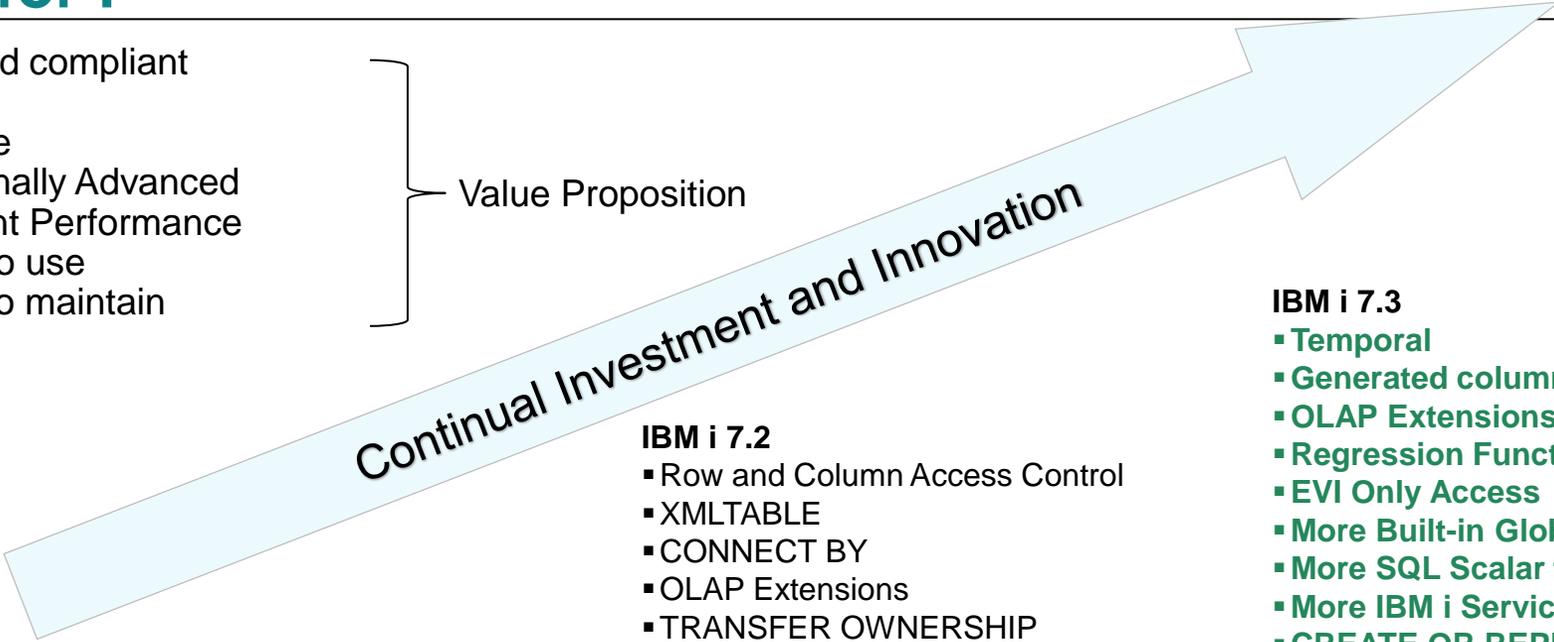
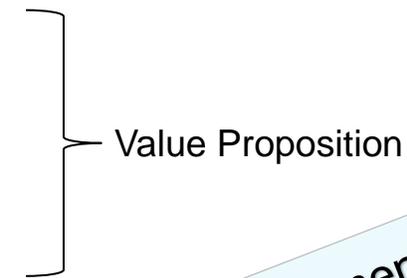
What's New with DB2 for i

7.3 +

Doug Mack
DB2 for i Analytics Consultant
IBM Lab Services
MACKD@us.ibm.com



- Standard compliant
- Secure
- Scalable
- Functionally Advanced
- Excellent Performance
- Easier to use
- Easier to maintain



IBM i 7.1

- XML Support
- Encryption enhancements (FIELDPROCs)
- Result set support in embedded SQL
- CURRENTLY COMMITTED
- MERGE
- Global variables
- Array support in procedures
- Three-part names and aliases
- SQE Logical file support
- EVI enhancements
- Inline functions
- CREATE OR REPLACE
- DECLARE GLOBAL TEMPORARY TABLE
- Catalog views

IBM i 7.2

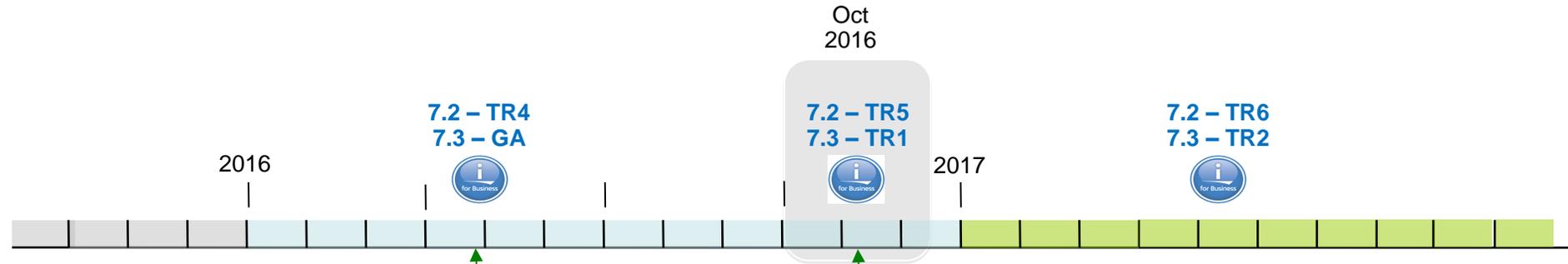
- Row and Column Access Control
- XMLTABLE
- CONNECT BY
- OLAP Extensions
- TRANSFER OWNERSHIP
- Named arguments and defaults for parameters
- Obfuscation of SQL routines
- Array support in UDFs
- Timestamp precision
- Multiple-action Triggers
- Built-in Global Variables
- 1.7 Terabyte Indexes
- Navigator Graphing and Charting
- System Limits for IFS

IBM i 7.3

- Temporal
- Generated columns for auditing
- OLAP Extensions
- Regression Functions / Covariance
- EVI Only Access
- More Built-in Global Variables
- More SQL Scalar functions
- More IBM i Services
- CREATE OR REPLACE TABLE
- ATTACH & DETACH PARTITION
- DBE tooling and automation
- High priority customer requests

| TR | SQL Enhancement |
|----------|-------------------------|
| TR5/TR1 | JSON_TABLE() |
| TR4 | Fair Lock / No Lock |
| TR11/TR3 | LIMIT & OFFSET |
| TR10/TR2 | CREATE OR REPLACE TABLE |
| TR9/TR1 | Regular Expressions |

DB2 for i – Enhancements delivered via DB2 PTF Groups



Enhancements timed with TR4

- Inlined UDTFs
- Trigger (re)deployment
- More IBM i Services
- New DB2 built-in Global Variables
- Enhanced SQL Scalar functions
- Evaluation option for DB2 SMP & DB2 Multisystem

Enhancements in 7.3:

- Temporal Tables
- Generated columns for auditing
- New OLAP built-ins
- Raised architecture limits
- New support for partitioned tables
- More IBM i Services
- And much more

Enhancements timed with TR5 & TR1

- **JSON_TABLE()**
- **INCLUDE for SQL Routines**
- **Database features in ACS**
- **Faster Scalar Functions**
- **More IBM i Services**
- **New DB2 for i Services**
- **And much more...**

www.ibm.com/developerworks/ibmi/techupdates/db2

IBM i 7.3 & DB2 for i

DB2 for i and IBM i 7.3

Major enhancements in DB2 for i deliver significant client value:

- **Temporal Tables – History of rows**
Data-centric, easily deployed, robust SQL point-in-time capability
- **Online Analytical Processing (OLAP) built-in functions**
Adding more analytics capabilities directly into DB2 for I
- **Generated Columns for auditing – Row level identity**
Let DB2 maintain the who, what, & how a row came to be
- **New DB2 Web Query Version 2.2**
Heterogeneous Database Access for Reporting or ETL
- **And... all the TR-timed enhancements delivered to IBM i 7.2**
IBM i Services, DB2 JSON Store, LIMIT and OFFSET, and many more



DB2 for i & Temporal

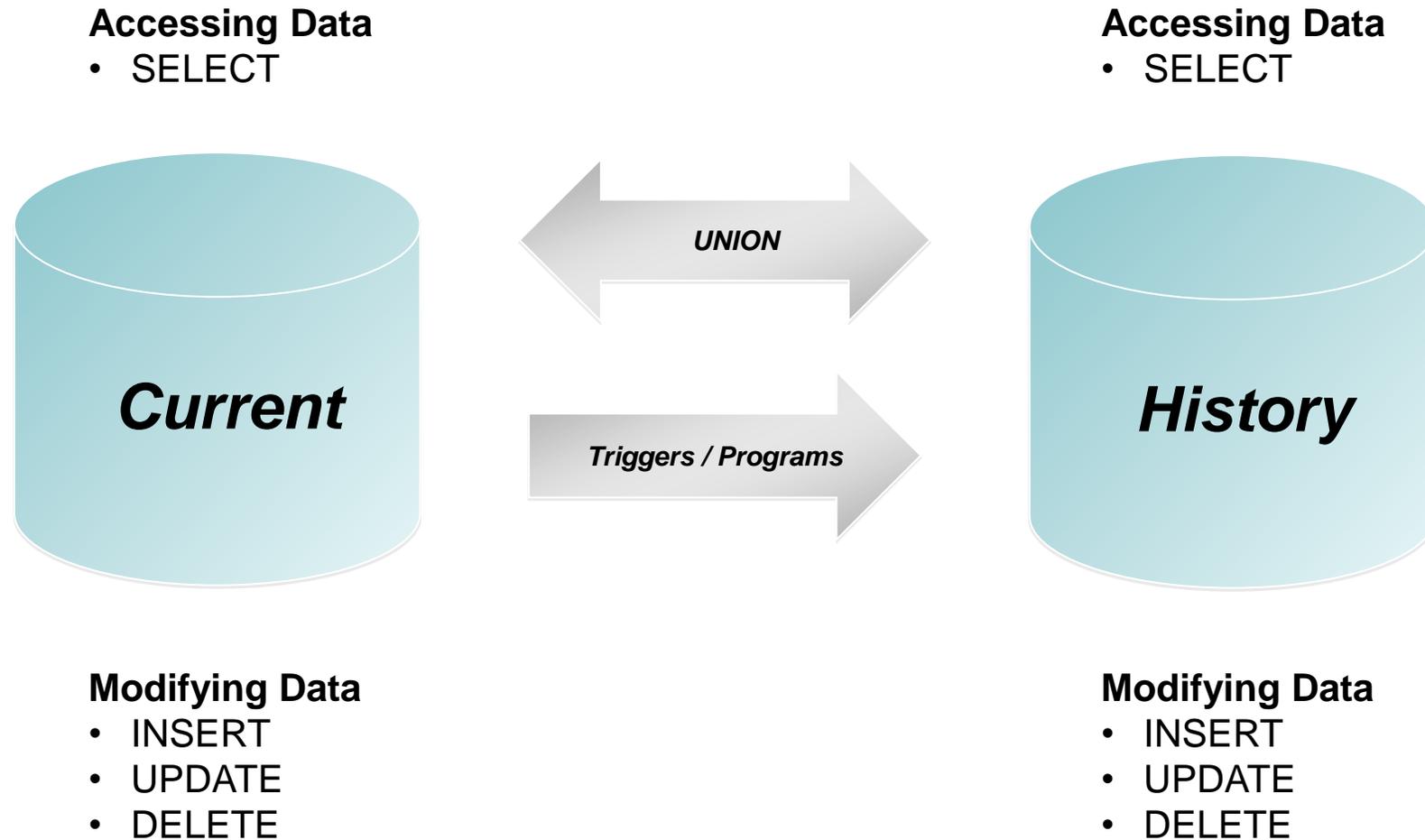
DB2 for i and Temporal Tables

With Temporal Tables, you can answer time-based questions:

- **Who was the client rep as of two years ago?**
- **Who were the client reps over the last five years?**
- **Produce an inventory report using a different point in time**

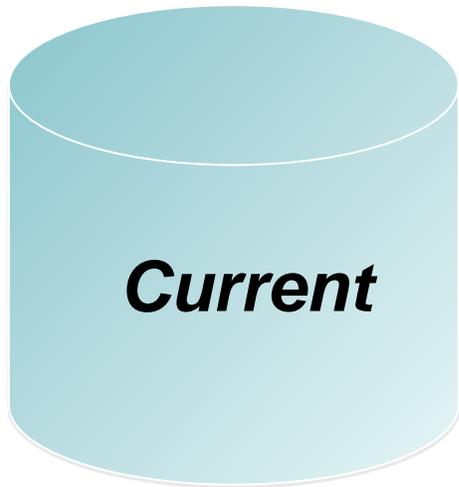


DB2 for i – The Old Way: Do It Yourself



DB2 for i – Another Way: Maintain History in a Data Warehouse

Accessing Data
 • SELECT

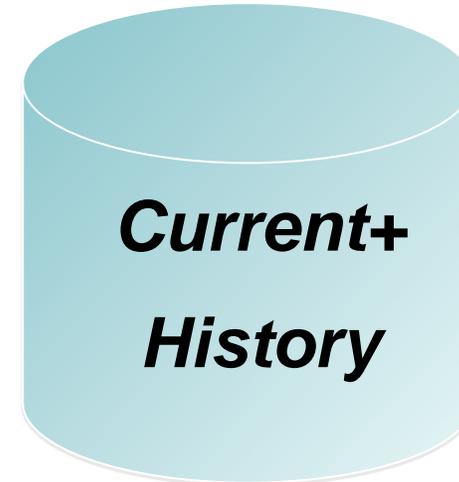


Modifying Data
 • INSERT
 • UPDATE
 • DELETE



Modifying Data
 • INSERT
 • UPDATE
 • DELETE

Accessing Data
 • SELECT



ETL tool automates handling of history data in data model



| <u>C_PKey</u> | <u>Customer_No</u> | Name | Address | City | State | Current Flag | <u>Begin Date</u> | <u>End Date</u> |
|---------------|--------------------|-----------|--------------------------|------------|------------|--------------|-------------------|-----------------|
| 00001 | A12345 | Doug Mack | 780 1 st Ave | Hagerstown | Maryland | 0 | 1976/01/01 | 1980/08/31 |
| 00002 | A12345 | Doug Mack | 4821 5 th Ave | Rochester | Minnesota | 0 | 1980/09/01 | 1995/05/31 |
| 00003 | A12345 | Doug Mack | 8568 Bainbridge | Olympia | Washington | 1 | 1995/06/01 | 9999/12/31 |

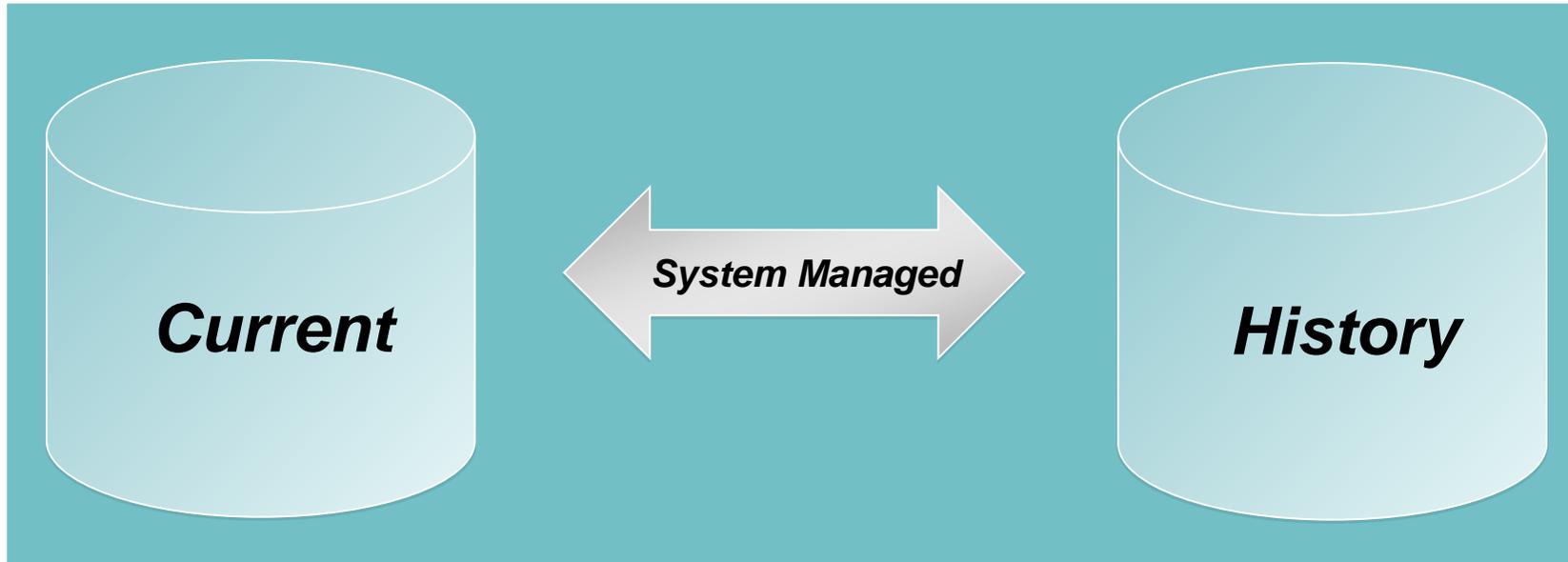
What's New: DB2 Temporal Tables

Accessing Data

- SELECT

Accessing Data

- SELECT



Modifying Data

- INSERT
- UPDATE
- DELETE

Modifying Data

- ~~INSERT~~
- ~~UPDATE~~
- DELETE (DBE Only)

Accessing a Temporal Table

- **SQL statements just reference the current table**
- **New clauses on the SELECT statement**
 - FOR SYSTEM TIME AS OF <value>
 - FOR SYSTEM TIME FROM <value> TO <value>
 - FOR SYSTEM TIME BETWEEN <value> AND <value>
- **New special register**
 - CURRENT TEMPORAL SYSTEM_TIME



Temporal construction for data-centric history

```
ALTER TABLE account
  ADD COLUMN row_birth
    TIMESTAMP(12) NOT NULL IMPLICITLY HIDDEN
    GENERATED ALWAYS AS ROW BEGIN
  ADD COLUMN row_death
    TIMESTAMP(12) NOT NULL IMPLICITLY HIDDEN
    GENERATED ALWAYS AS ROW END
  ADD COLUMN transaction_time
    TIMESTAMP(12) IMPLICITLY HIDDEN
    GENERATED ALWAYS AS TRANSACTION START ID
  ADD PERIOD SYSTEM_TIME (row_birth, row_death)
```

Establish birth/death of a row

```
CREATE TABLE account_hist LIKE account
```

Create history table

```
ALTER TABLE account
  ADD VERSIONING USE HISTORY TABLE account_hist
```

Enable Temporal tracking

Move Work to DB2 for i Example: Temporal Tables (7.3)

With DB2 Temporal Tables, you can:

- **Who was the client rep as of two years ago?**
SELECT CLIENT_REP FROM ACCOUNTS
FOR SYSTEM_TIME AS OF CURRENT_TIMESTAMP – 2 YEARS
- **Who were the client reps over the last five years?**
SELECT CLIENT_REP FROM ACCOUNTS
FOR SYSTEM_TIME FROM CURRENT_TIMESTAMP – 5 YEARS
TO CURRENT_TIMESTAMP
- **Produce an inventory report using a different point in time**
SET **CURRENT TEMPORAL SYSTEM_TIME** '2016-03-22 17:00:00';
CALL GENERATE_INVENTORY_REPORT();



For more details, attend session at 11:50 on Thursday

DB2 for i & OLAP

SQL OLAP Specification Extensions

On-Line Analytical Processing (OLAP) specifications provide the ability to return ranking, row numbering, and other aggregate function information as a scalar value in a query result. RANK, DENSE_RANK, and ROW_NUMBER were added in IBM i 6.1.

```
SELECT workdept, lastname, salary,
       RANK() OVER(PARTITION BY workdept ORDER BY salary DESC) rank,
       DENSE_RANK() OVER(PARTITION BY workdept ORDER BY salary DESC) dense_rank,
       ROW_NUMBER() OVER(ORDER BY workdept, lastname) row_number
FROM employee ORDER BY workdept, lastname
```

| WORKDEPT | LASTNAME | SALARY | RANK | DENSE_RANK | ROWNBR |
|----------|-----------|-----------|------|------------|--------|
| AOO | HAAS | 52,750.00 | 1 | 1 | 1 |
| AOO | HEMMINGER | 46,500.00 | 2 | 2 | 2 |
| AOO | LUCCHESSI | 46,500.00 | 2 | 2 | 3 |
| AOO | O'CONNELL | 29,250.00 | 4 | 3 | 4 |
| AOO | ORLANDO | 29,250.00 | 4 | 3 | 5 |

SQL Query Engine and OLAP Specification Extensions

With IBM i 7.3, DB2 for i supports the following additional OLAP specifications for summarizing and referencing rows **within windows** of result sets:

- **Aggregates (SUM, COUNT, etc.)**
- **LAG and LEAD**
- **NTILE**
- **FIRST_VALUE, NTH_VALUE, and LAST_VALUE**
- **RATIO_TO REPORT**
- **CUME_DIST**

What is the value of this?

- **You could build analytical applications with much less complexity**
- **BI tools like DB2 Web Query and Cognos can “push the work” to DB2 for better performance**

Scalar Aggregate Functions SUM and PARTITION BY

Return the detail store information and the total sales by region plus the percentage the store contributed to the total for the region:

```
SELECT store, region, sales,
       SUM(sales) OVER(PARTITION BY region) region_total,
       DECIMAL(100*sales / SUM (sales) OVER(PARTITION BY region), 5,2) percentage
FROM stores ORDER BY region, percentage
```

| STORE | REGION | SALES | REGION TOTAL | PERCENTAGE |
|-------|--------|------------|--------------|------------|
| Wally | NE | 150,000.00 | 450,000.00 | 33.33 |
| Pensk | NE | 300,000.00 | 450,000.00 | 66.66 |
| Bobs | NW | 100,000.00 | 1,040,000.00 | 9.61 |
| Toms | NW | 440,000.00 | 1,040,000.00 | 42.30 |
| Mills | NW | 500,000.00 | 1,040,000.00 | 48.07 |

Window 1

Window 2

Rolling Sum Scalar Aggregate OLAP example

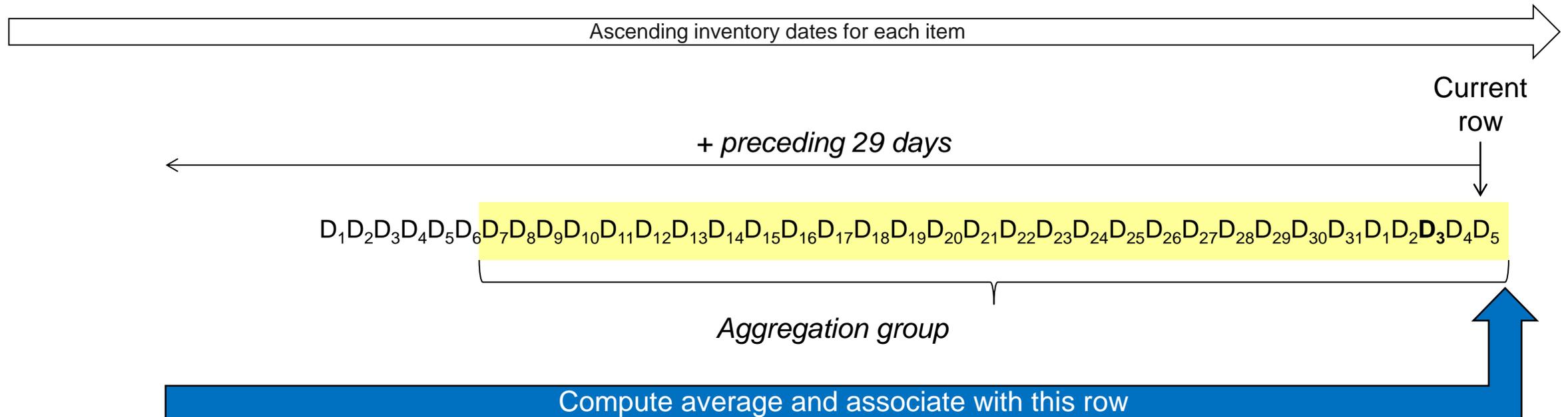
Return the detail store information and the rolling sum of the store sales:

```
SELECT store, region, sales,
       SUM(sales) OVER(ORDER BY sales DESC) rolling_sum
FROM stores ORDER BY rolling_sum
```

| STORE | REGION | SALES | ROLLING_SUM |
|---------|--------|------------|--------------|
| Caining | SW | 770,000.00 | 770,000.00 |
| Mills | NW | 500,000.00 | 1,270,000.00 |
| Toms | NW | 440,000.00 | 1,710,000.00 |
| Menes | SW | 400,000.00 | 2,110,000.00 |
| BBB | SE | 350,000.00 | 2,460,000.00 |

OLAP Specifications - Aggregate

- Given daily inventory levels for each item, **compute a 30 day running average** for each day for a specified item
- Assume inventory levels are measured every day resulting in a row for each item, every day of the year (i.e. there are no missing dates)
- Use a window aggregation group specification to specify the 30 day moving window



OLAP Specifications - Aggregate

 OLAP Specification

WITH CTE (ITEM_NO, INVENTORY_DATE, MOVING_30_DAY_AVG) AS

```
(
  SELECT  D.ITEM_NO,
          D.INVENTORY_DATE,
          AVG(D.DAILY_QUANTITY) OVER (
              PARTITION BY D.ITEM_NO
              ORDER BY D.INVENTORY_DATE
              ROWS BETWEEN 29 PRECEDING AND CURRENT ROW)
          AS MOVING_30_DAY_AVG
  FROM    DAILY_INVENTORY D
  WHERE   D.ITEM_NO = 673091
  AND     D.INVENTORY_DATE BETWEEN DATE('2015-01-01') - 2 MONTHS AND '2015-12-31'
```

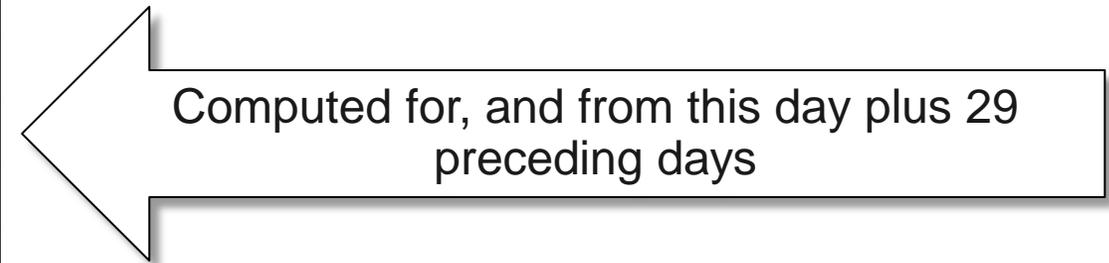
Compute average

```
)
SELECT  C.ITEM_NO,
        C.INVENTORY_DATE,
        C.MOVING_30_DAY_AVG
FROM    CTE C
WHERE   C.INVENTORY_DATE BETWEEN '2015-01-01' AND '2015-12-31'
ORDER BY C.ITEM_NO,
        C.INVENTORY_DATE;
```

Select specific averages

DB2 Web Query Report Built over SQL View Containing OLAP Function

| ITEM NUMBER | INVENTORY DATE | MOVING 30 DAY AVERAGE |
|-------------|-------------------|-----------------------|
| 673091 | 2015-02-01 | 17.00 |
| 673091 | 2015-02-02 | 13.25 |
| 673091 | 2015-02-03 | 11.40 |
| 673091 | 2015-02-04 | 12.33 |
| 673091 | 2015-02-05 | 14.71 |
| 673091 | 2015-02-06 | 13.75 |
| 673091 | 2015-02-07 | 16.11 |
| 673091 | 2015-02-08 | 15.90 |
| 673091 | 2015-02-09 | 15.27 |
| 673091 | 2015-02-10 | 14.58 |
| 673091 | 2015-02-11 | 16.23 |
| 673091 | 2015-02-12 | 16.85 |
| ... | ... | ... |



Computed for, and from this day plus 29 preceding days

First_Value, Last_Value, Nth_Value – OLAP Aggregate functions

Compare the sales of the current store to the store with the best sales, second best sales, and the worst sales results:

```
SELECT store, sales,
       sales - FIRST_VALUE(sales) OVER (ORDER BY sales DESC RANGE BETWEEN
         UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) behind_1st,
       sales - NTH_VALUE(sales,2) OVER (ORDER BY sales DESC RANGE BETWEEN
         UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) behind_2nd,
       sales - LAST_VALUE(sales) OVER (ORDER BY sales DESC RANGE BETWEEN
         UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) compared_to_last
FROM stores ORDER BY sales DESC
```

| STORE | SALES | BEHIND_1ST | BEHIND_2ND | COMPARED_TO_LAST |
|---------|------------|-------------|-------------|------------------|
| Caining | 770,000.00 | 0.00 | 270,000.00 | 670,000.00 |
| Mills | 500,000.00 | -270,000.00 | 0.00 | 400,000.00 |
| .. | ... | ... | ... | ... |
| Bobs | 100,000.00 | -670,000.00 | -400,000.00 | 0.00 |

Lag and Lead – Ordered OLAP specifications

Compare the sales of stores within the same region including comparisons to the stores that were adjacent in terms of better and worse sales:

```
SELECT store, region, sales,
       sales - LAG(sales,1) OVER(PARTITION BY region ORDER BY sales) AS prior_diff,
       LEAD(sales,1) OVER(PARTITION BY region ORDER BY sales) - sales AS next_diff
FROM stores ORDER BY region, sales
```

| STORE | REGION | SALES | PRIOR_DIFF | NEXT_DIFF |
|---------|--------|------------|------------|------------|
| Bobs | NW | 100,000.00 | - | 340,000.00 |
| Toms | NW | 440,000.00 | 340,000.00 | 60,000.00 |
| Mills | NW | 500,000.00 | 60,000.00 | - |
| Targe | SW | 140,000.00 | - | 260,000.00 |
| Menes | SW | 400,000.00 | 260,000.00 | 370,000.00 |
| Caining | SW | 770,000.00 | 370,000.00 | - |

NTILE and Ratio_To_Report – Ordered OLAP specification and aggregate function

Calculate the quartile ranking for all stores and show their overall sales percentage within the region:

```
SELECT store, region, sales,
       NTILE(4) OVER(ORDER BY sales DESC) quartile_rank,
       DECIMAL(RATIO_TO_REPORT(sales) OVER() *100, 10, 2) percent
FROM stores ORDER BY sales DESC
```

| STORE | REGION | SALES | QUARTILE_RANK | PERCENT |
|---------|--------|------------|---------------|---------|
| Caining | SW | 770,000.00 | 1 | 21.10 |
| Mills | NW | 500,000.00 | 1 | 13.70 |
| Toms | NW | 440,000.00 | 1 | 12.05 |
| Menes | SW | 400,000.00 | 2 | 10.96 |
| ... | | | ... | ... |
| Bobs | NW | 100,000.00 | 4 | 2.74 |

SQL OLAP Specification Extensions for Statistics

With IBM i 7.3, DB2 for i also supports the following OLAP specifications for **statistical analysis**:

- **Correlation**
- **Covariance**
- **Percentile**
- **Median**
- **Linear regression (slope, intercept, etc.)**

https://en.wikipedia.org/wiki/Regression_analysis

Regression analysis is a statistical process for estimating the relationships among variables.

Correlation, Covariance and Covariance_Samp Aggregate function

Use correlation and covariance to analyze the relationship between salary and bonus for each department:

```
SELECT workdept,
       CORRELATION(salary, bonus) correlation,
       COVARIANCE(salary, bonus) covariance,
       COVARIANCE_SAMP(salary, bonus) covariance_samp
FROM employee GROUP BY workdept ORDER BY workdept
```

| WORKDEPT | CORRELATION | COVARIANCE | COVARIANCE_SAMP |
|----------|-------------|------------|-----------------|
| A00 | 0.976023 | 1,743,000 | 2,178,750 |
| B01 | - | 0 | - |
| C01 | 0.999835 | 574,437 | 765,916 |
| D11 | 0.775424 | 240,454 | 264,500 |
| ... | | | ... |
| E21 | 0.910221 | 68,944 | 82,733 |

Correlation, Covariance and Covariance_Samp OLAP functions

Use the correlation and covariance aggregate functions in an OLAP expression to further analyze for department 'A00' the relationship between salary and bonus:

```
SELECT empno,
  CORRELATION(salary, bonus) OVER(PARTITION BY workdept ORDER BY empno) correlation,
  COVARIANCE(salary, bonus) OVER(PARTITION BY workdept ORDER BY empno) covariance
FROM employee WHERE workdept = 'A00' ORDER BY empno
```

| EMPNO | CORRELATION | COVARIANCE |
|--------|-------------|------------|
| 000010 | - | 0 |
| 000110 | 1.000000 | 156,250 |
| 000120 | 0.999853 | 1,688,888 |
| 200010 | 0.962723 | 1,381,250 |
| 200120 | 0.976023 | 1,743,000 |

Percentile Continuous, Percentile Discrete, and Median Aggregate functions

Use the Percentile_Cont, Percentile_Disc and Median aggregate functions to analyze the relationship between employee salaries and bonuses:

```
SELECT workdept,
       PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY salary) percentile_cont,
       PERCENTILE_DISC(0.5) WITHIN GROUP(ORDER BY salary) percentile_disc,
       MEDIAN(bonus) median
FROM employee GROUP BY workdept
```

| WORKDEPT | PERCENTILE_CONT | PERCENTILE_DISC | MEDIAN |
|----------|-----------------|-----------------|--------|
| A00 | 46,500.00 | 46,500.00 | 900.00 |
| B01 | 1,250.00 | 1,250.00 | 800.00 |
| C01 | 28,420.00 | 28,420.00 | 600.00 |
| D11 | 24,680.00 | 24,680.00 | 500.00 |
| ... | ... | ... | ... |
| E21 | 24,605.00 | 23,840.00 | 500.00 |

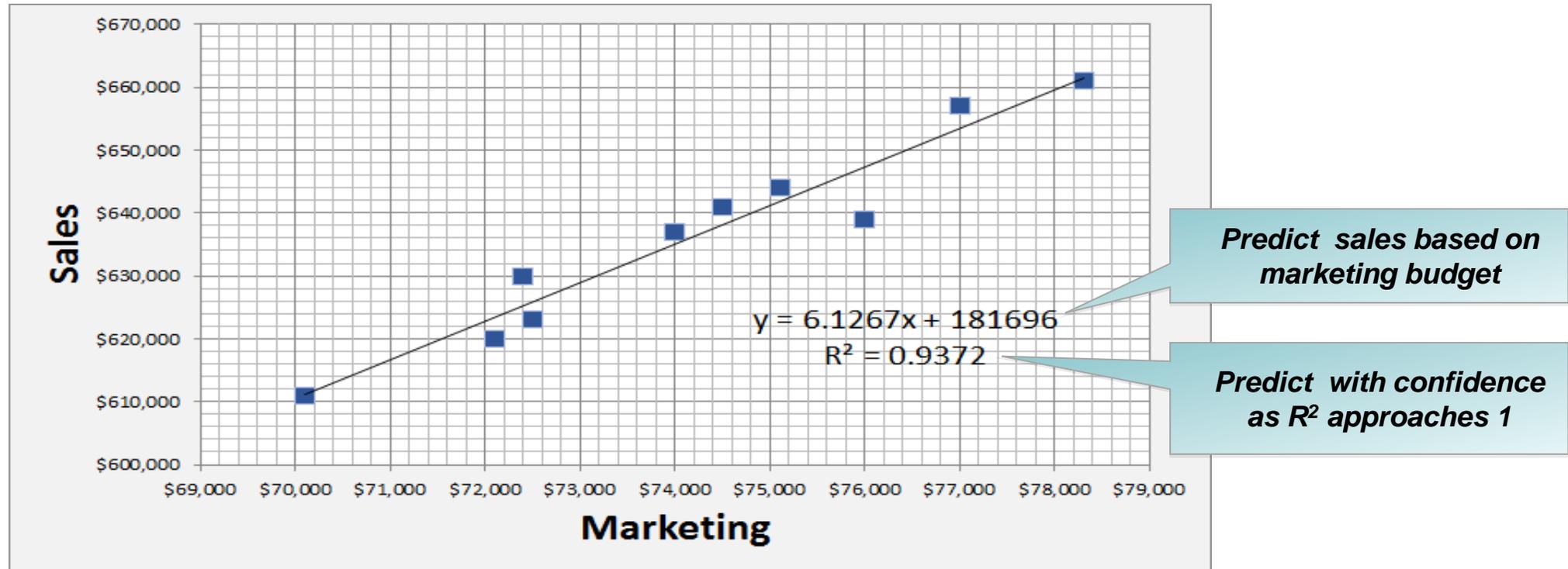
Regression Aggregate Functions

Business questions:

- Is there a correlation between the amount spent on marketing and sales for a product?
- Is the correlation weak or strong?
- Can we predict sales based on the amount spent on marketing?

| Year/Quarter | Marketing | Sales |
|--------------|-----------|-----------|
| 2014 Q1 | \$70,100 | \$611,000 |
| 2014 Q2 | \$77,000 | \$657,000 |
| 2014 Q3 | \$72,100 | \$620,000 |
| 2014 Q4 | \$72,500 | \$623,000 |
| 2015 Q1 | \$78,300 | \$661,000 |
| 2015 Q2 | \$74,500 | \$641,000 |
| 2015 Q3 | \$74,000 | \$637,000 |
| 2015 Q4 | \$72,400 | \$630,000 |
| 2016 Q1 | \$75,100 | \$644,000 |
| 2016 Q2 | \$76,000 | \$639,000 |

Regression Aggregate Functions



Business results:

- SELECT **REGR_SLOPE** (sales, mktg), **REGR_INTERCEPT** (sales, mktg)
FROM salesdata
- SELECT POWER (**CORRELATION** (sales, mktg), 2)
FROM salesdata

DB2 for i & Row Level Auditing

Enhanced data-centric auditing – with autogenerated columns

- **Auto-generated columns are a very powerful building block for datacentric programming in that they direct the database to automatically generate column values for you.**
- **Prior to IBM i 7.3, DB2 for i supported:**
 - IDENTITY columns (which are very good for surrogate primary keys)
 - ROW CHANGE TIMESTAMP (which records the time whenever a row is changed)
- **The SQL syntax GENERATED ALWAYS prevents anyone from modifying those column values, including a knowledgeable hacker.**
- **IBM i 7.3 includes support for additional options:**
 - DATA CHANGE OPERATION (I/U/D)
 - Special register
 - Built-in Global Variable

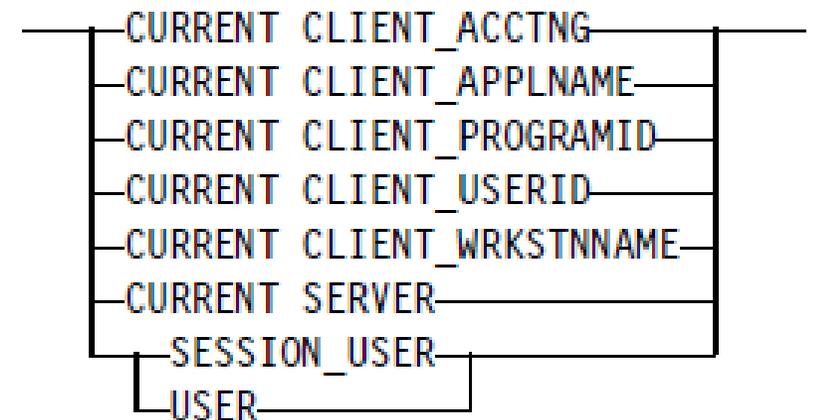
Autogenerated columns – DATA CHANGE OPERATION

- **DATA CHANGE OPERATION** is a one character value recording the last data change:
 - **I = Insert**
 - **U = Update**
 - **D = Delete**
- **These work well with temporal tables in that history table will provide a timeline of what changes were made and when.**
 - **The Delete record will be included if the temporal table was configured with the ON DELETE ADD EXTRA ROW clause.**

```
ALTER TABLE fact_table
  ADD COLUMN audit_type_change CHAR (1)
  GENERATED ALWAYS AS (DATA CHANGE OPERATION)
```

Autogenerated columns – special registers

- **Special registers can be used to record information about the user making the change and/or the application environment.**
- **Client registers can be set by the application:**
 - Web users might access IBM i data using a generic user profile of **WEBUSER**.
 - The application can use the **CURRENT CLIENT_USERID** special register to record that the user name at the web interface is **JIM**.
- **SESSION_USER** and **USER** contain the user profile currently in use which identifies who is making a change to the database.



```
ALTER TABLE fact_table
  ADD COLUMN audit_app_client_userid VARCHAR(255)
    GENERATED ALWAYS AS (CURRENT CLIENT_USERID)
  ADD COLUMN audit_user VARCHAR(128)
    GENERATED ALWAYS AS (SESSION_USER)
```

Autogenerated columns – built-in global variables

- **Built-in global variables are managed by the system and provide additional environmental information.**
- **You can use these to monitor things like which job or which IP address is being used to make a change to the database.**

```
ALTER TABLE fact_table
  ADD COLUMN audit_job_name VARCHAR(28)
    GENERATED ALWAYS AS (QSYS2.JOB_NAME)
  ADD COLUMN audit_client_IP VARCHAR(128)
    GENERATED ALWAYS AS (SYSIBM.CLIENT_IPADDR)
```

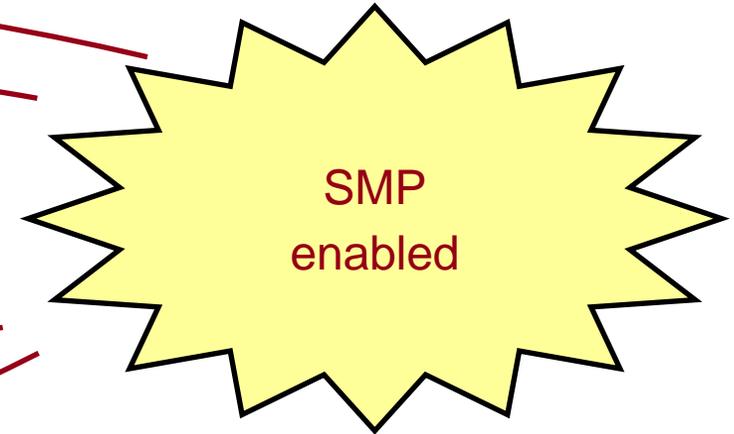
| |
|------------------------------|
| QSYS2.JOB_NAME |
| QSYS2.SERVER_MODE_JOB_NAME |
| SYSIBM.CLIENT_HOST |
| SYSIBM.CLIENT_IPADDR |
| SYSIBM.CLIENT_PORT |
| SYSIBM.PACKAGE_NAME |
| SYSIBM.PACKAGE_SCHEMA |
| SYSIBM.PACKAGE_VERSION |
| SYSIBM.ROUTINE_SCHEMA |
| SYSIBM.ROUTINE_SPECIFIC_NAME |
| SYSIBM.ROUTINE_TYPE |

DB2 for i

Limits & Scalability

DB2 Symmetric Multiprocessing (feature of IBM i)

- **SELECTING**
 - Index scan or probe
 - Table scan or probe via bitmap or RRN list
 - Table scan
- **JOINING**
 - Index scan or probe
 - Hash
- **GROUPING**
 - Index scan or probe
 - Hash
- **ORDERING**
 - Index scan or probe
 - Sort
- System Value: QQRVDEGREE controls SMP settings



- Creating temporary indexes for joining, grouping or ordering is SMP enabled

OLAP and DB2 SMP

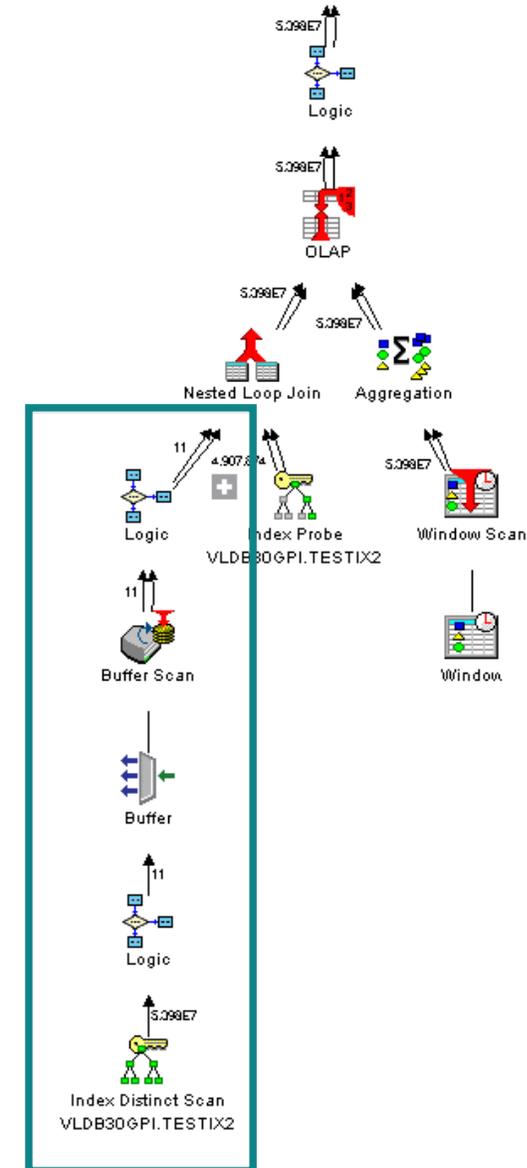
Prior to IBM i 7.3, the part of the query tree containing OLAP expressions could not leverage DB2 SMP for parallel execution

With 7.3, it is possible to evaluate the OLAP results in parallel.

- DB2 SMP must be enabled.
- The OLAP must contain a PARTITION BY clause
- The feeding tree may be implemented using either an index or sort.

```

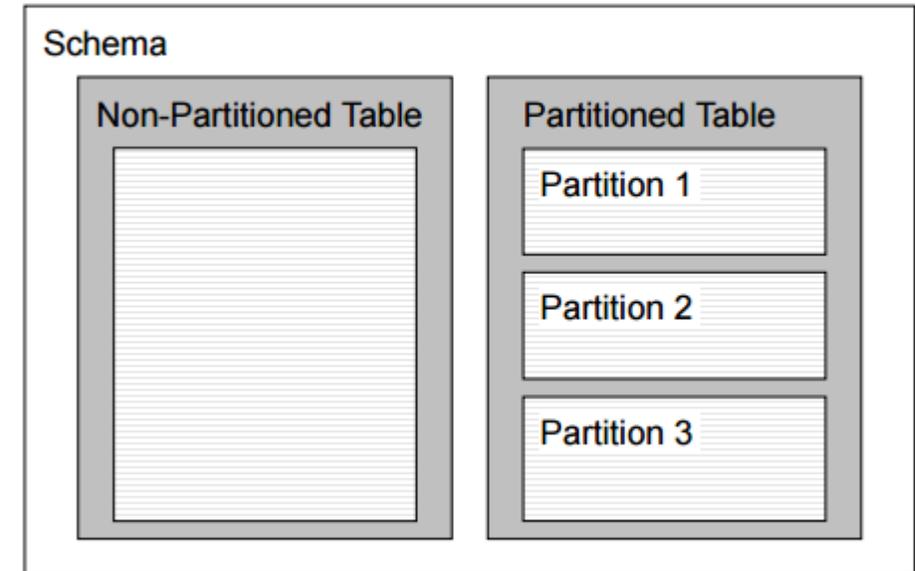
SELECT discount, quantity,
       COUNT(quantity) OVER(PARTITION
                           BY discount ORDER BY quantity)
FROM item_fact
OPTIMIZE FOR ALL ROWS
    
```



DB2 MultiSystem (feature of IBM i)

- Provides Concept of Partitioning of a Table
- Why
 - Extend table size and number of records beyond current limits
 - 42B record limit goes to 256 x 42B !!!!!!!
 - Management benefits
 - Want to keep 12 months of rolling data and lop off the oldest month when a new one comes on?
 - Can use simple DROP command
 - Might want to just backup current month if that is the only active partition for data updates
- Planning Required
 - See white paper on DB2 for i Wiki: <https://ibm.biz/Bd4fFb>
 - DB2 for i VLDB Consulting Workshop
 - Contact Mike Cain at mcain@us.ibm.com

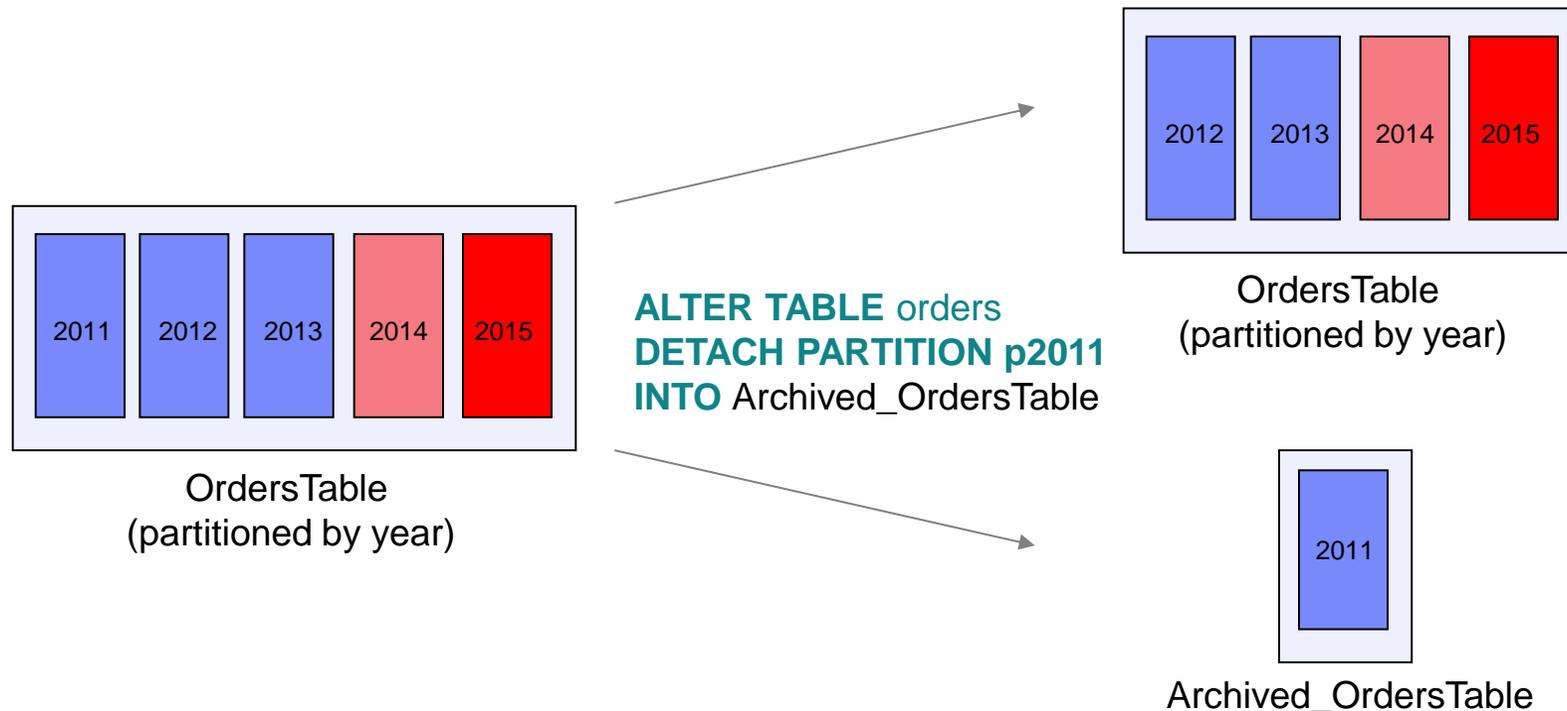
Partitioned Table Diagram



ALTER TABLE ATTACH and DETACH Partitions

ALTER TABLE DETACH PARTITION allows for the efficient roll-out of a partition that is no longer needed to be kept online.

- ❑ ALTER TABLE DROP PARTITION – Delete the data
- ❑ ALTER TABLE DETACH PARTITION – Retain the data, in a new single partition table



DB2 for i priced OS options – evaluation copy

Try before you buy!

DB2 Symmetric Multiprocessing – Option 26
DB2 Multisystem – Option 27

The DB2 for i Lab Services team has the ability to allow you to evaluate either of these options for up to 70 days, for no charge.

This is a simpler, no strings attached, way to evaluate these valuable database options.

Interested?

Contact

Rob Bestgen (bestgen@us.ibm.com) or
Scott Forstie (forstie@us.ibm.com)



Raised architectural limits in IBM i 7.3

Maximum # of objects in a Library

360,000 → 1,000,000

Maximum # of parameters in a DB2 procedure

1,024 → 2,000

Maximum # of parameters in a DB2 function

90 → 2,000

Maximum # of return columns from a DB2 table function

1024 → 8,000

Maximum # of members referenced in an SQL view

256 → 1,000



Encoded Vector Index Index Only Access (EOA)

- Using Bundesliga Football Example



Football & EOA

```
SELECT Email_Address  
FROM Bundesliga_Fans  
WHERE FCName = 'Eintracht'
```

Football & EOA

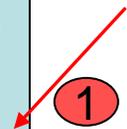
```
SELECT Email_Address
FROM Bundesliga_Fans
WHERE FCName = 'Eintracht'
```

EVI over FCName

Symbol Table

| Key Column | Count | cd |
|----------------------|------------|----|
| Wolfsburg | 1.000.000 | 1 |
| Dortmund | 8.500.000 | 2 |
| ... | ... | 3 |
| Bayern Munich | 26.700.000 | 4 |
| Eintracht | 4.000.003 | 5 |
| ... | | |

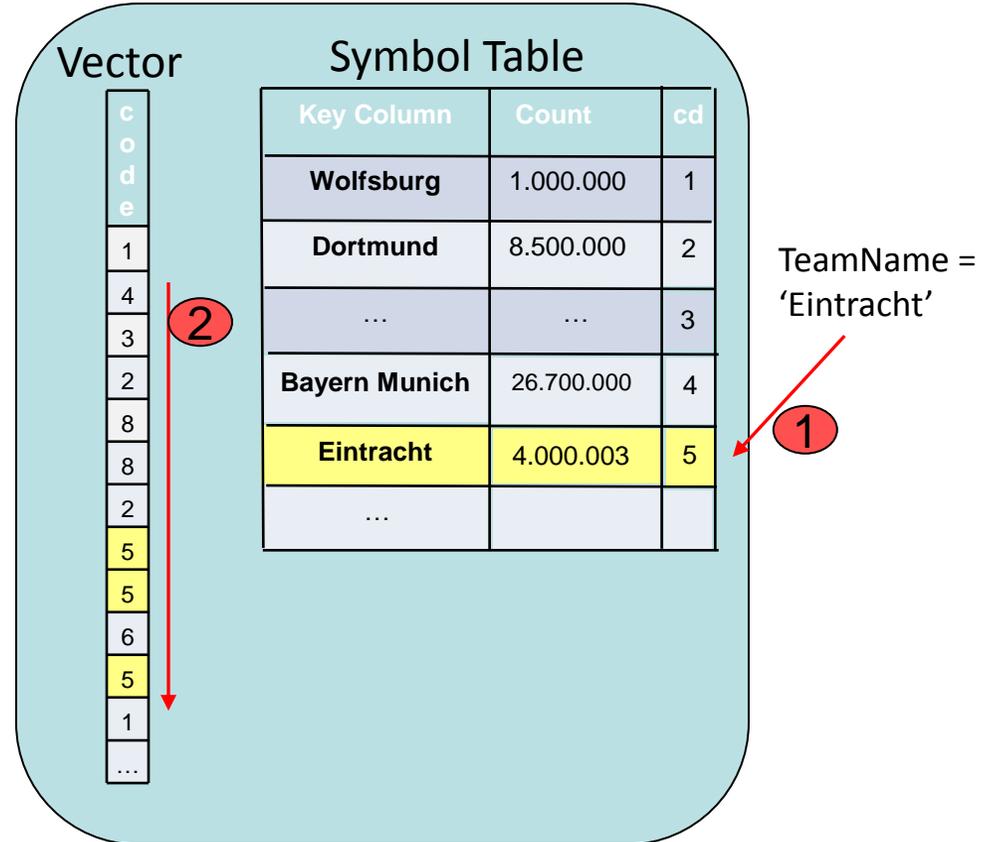
TeamName = 'Eintracht'



Football & EOA

```
SELECT Email_Address
FROM Bundesliga_Fans
WHERE FCName = 'Eintracht'
```

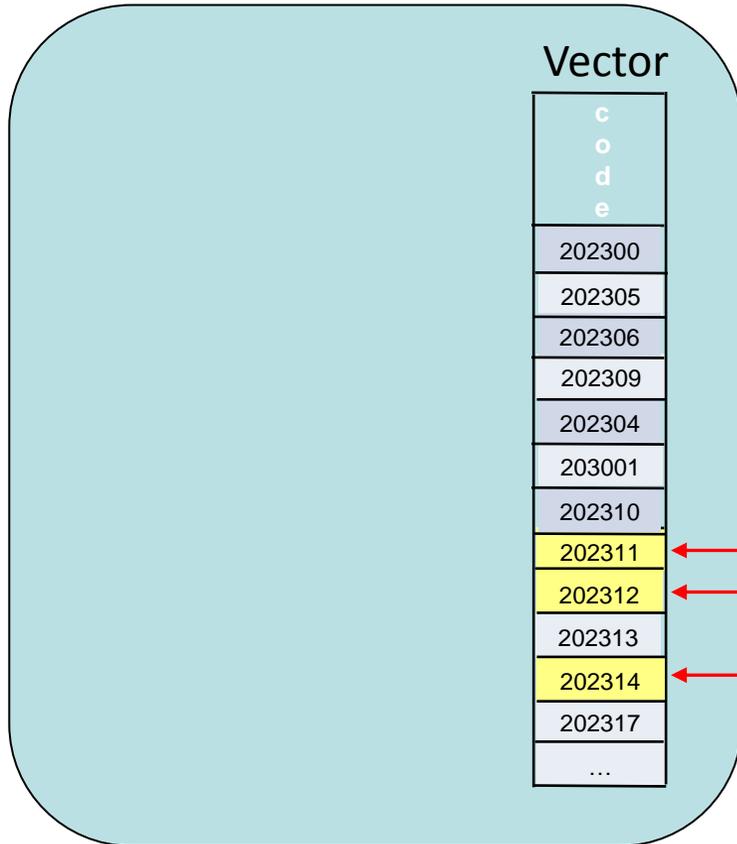
EVI over FCName



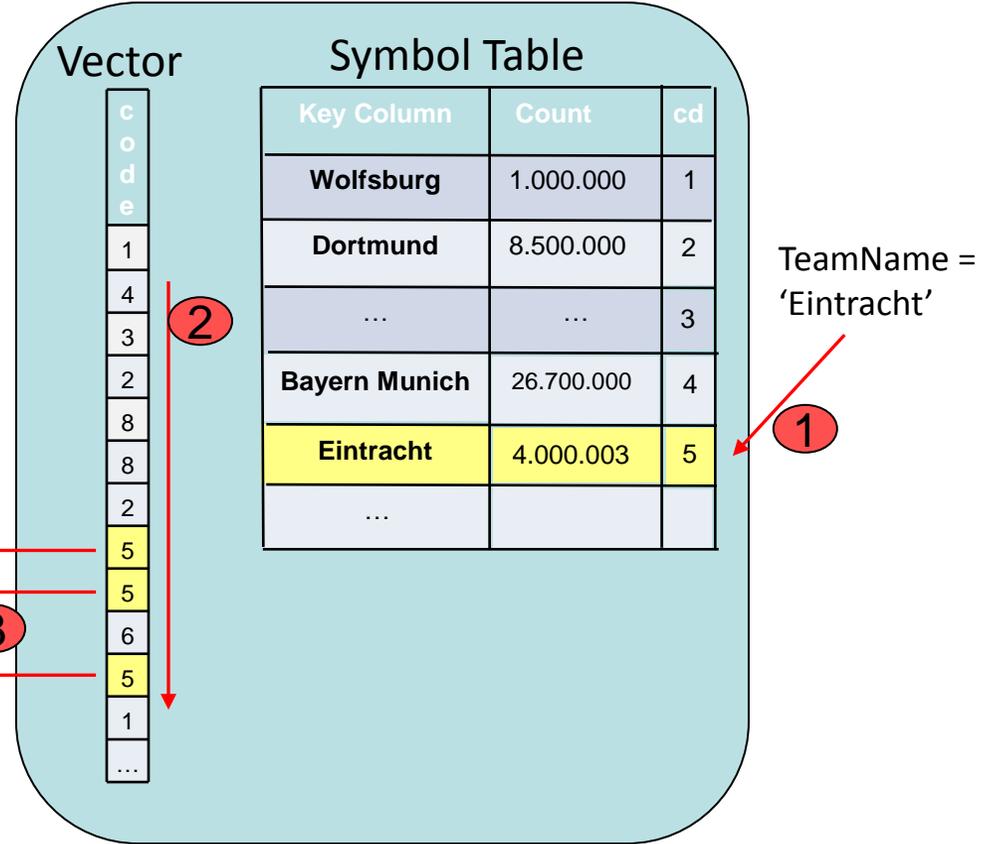
Football & EOA

```
SELECT Email_Address
FROM Bundesliga_Fans
WHERE FCName = 'Eintracht'
```

EVI over Email_Address



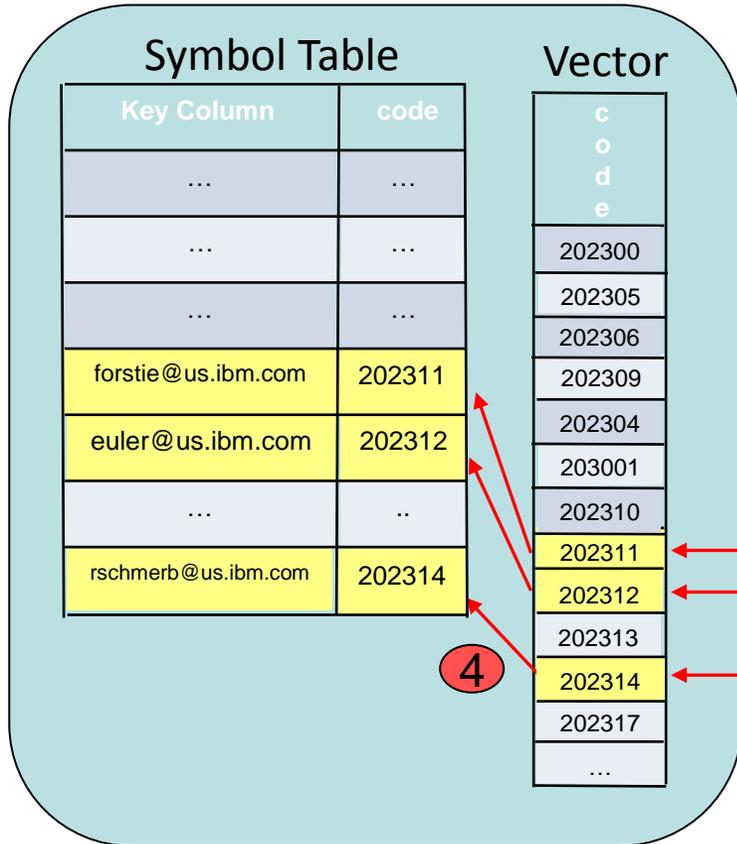
EVI over FCName



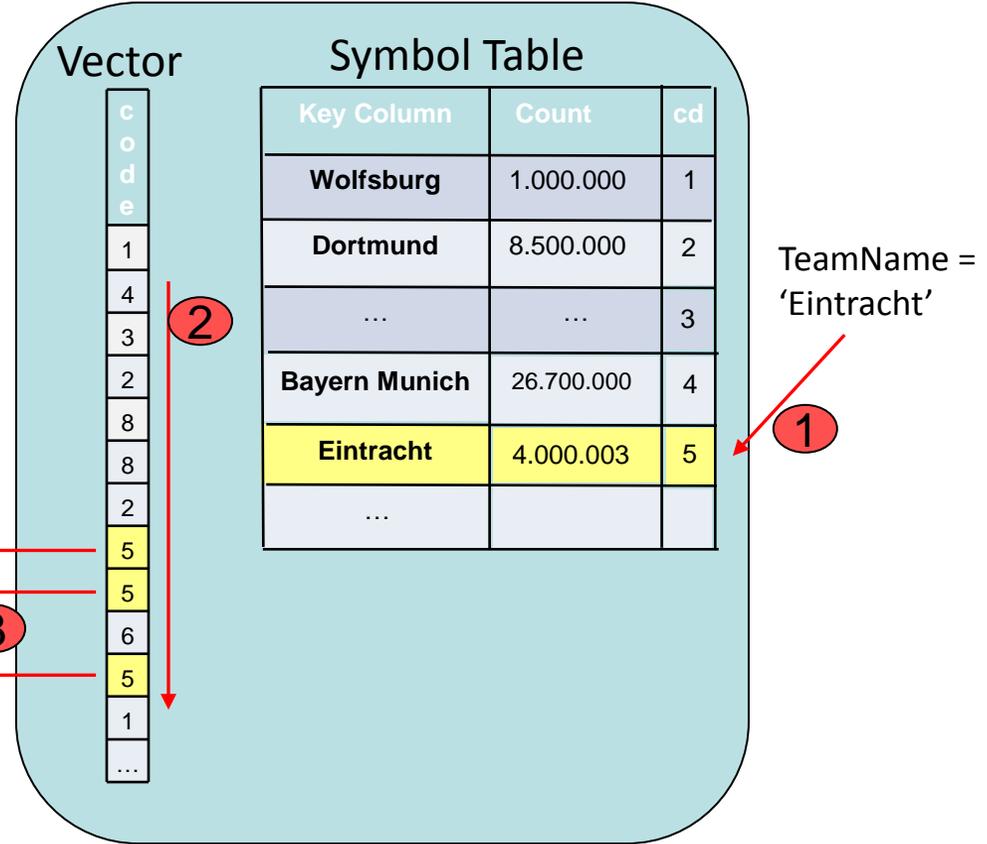
Football & EOA

```
SELECT Email_Address
FROM Bundesliga_Fans
WHERE FCName = 'Eintracht'
```

EVI over Email_Address



EVI over FCName



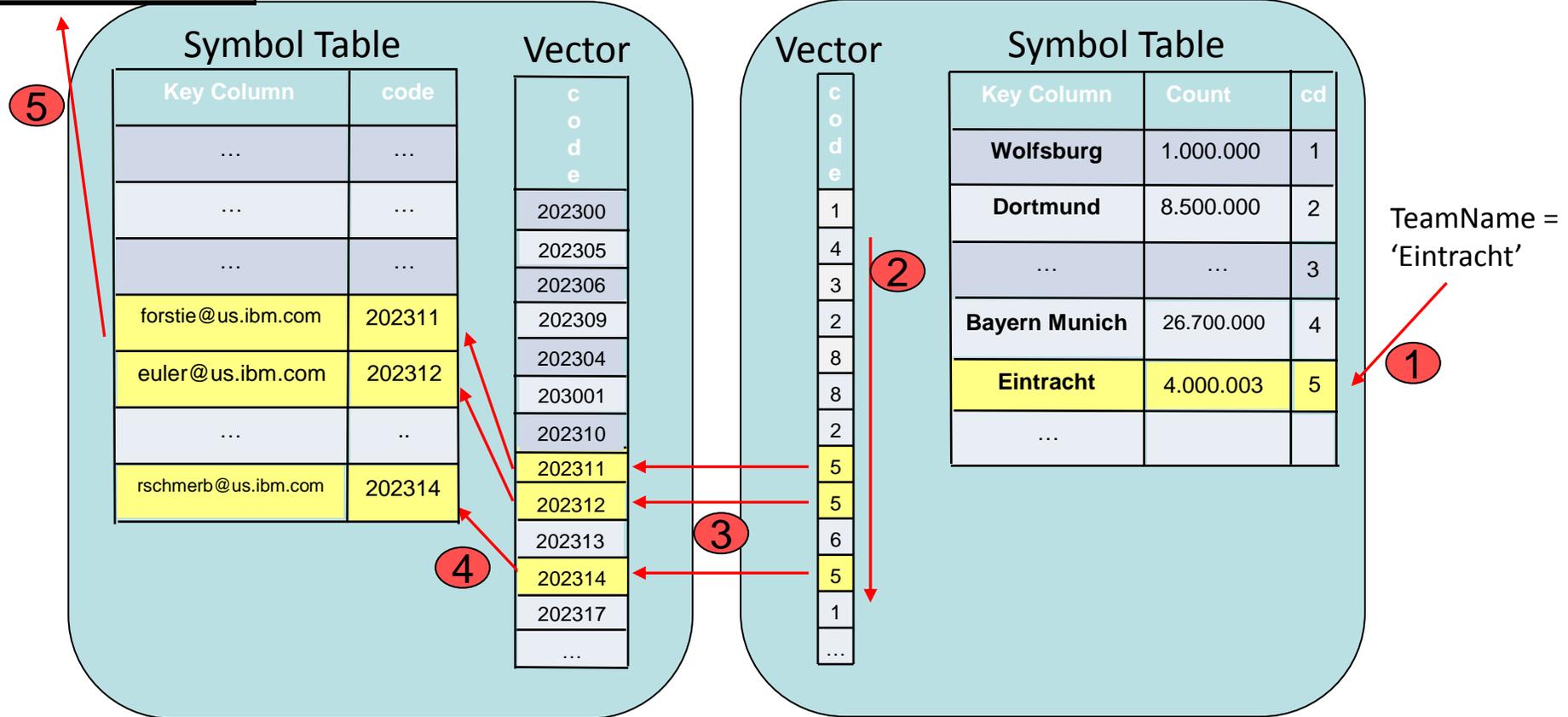
Football & EOA

```
SELECT Email_Address
FROM Bundesliga_Fans
WHERE FCName = 'Eintracht'
```

| |
|---------------------|
| forstie@us.ibm.com |
| euler@us.ibm.com |
| rschmerb@us.ibm.com |

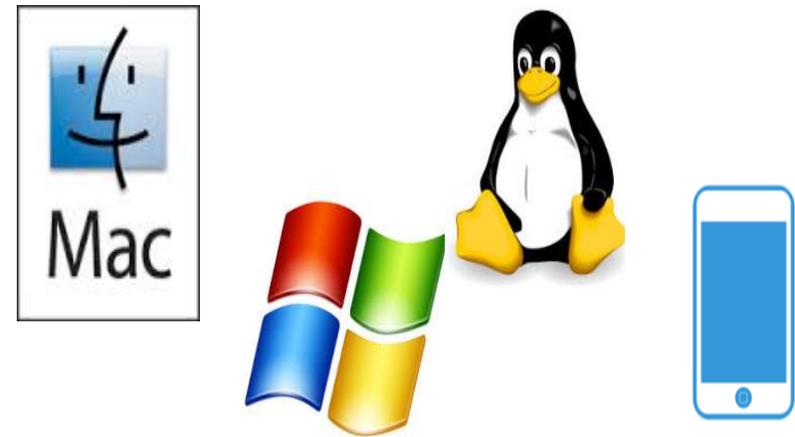
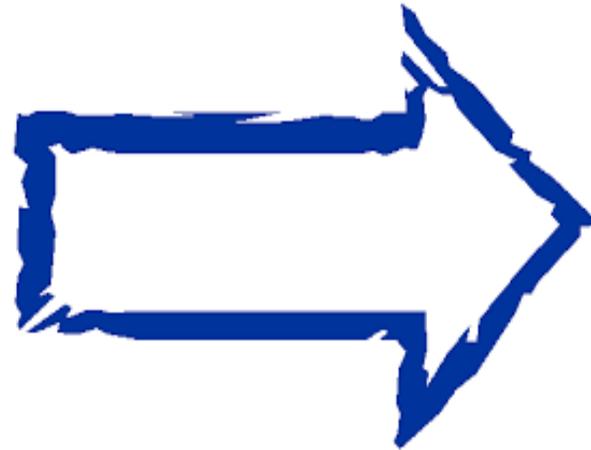
EVI over Email_Address

EVI over FCName



Navigator & DB2 for i

Strategy



Any Device

ACS – Plan for database features

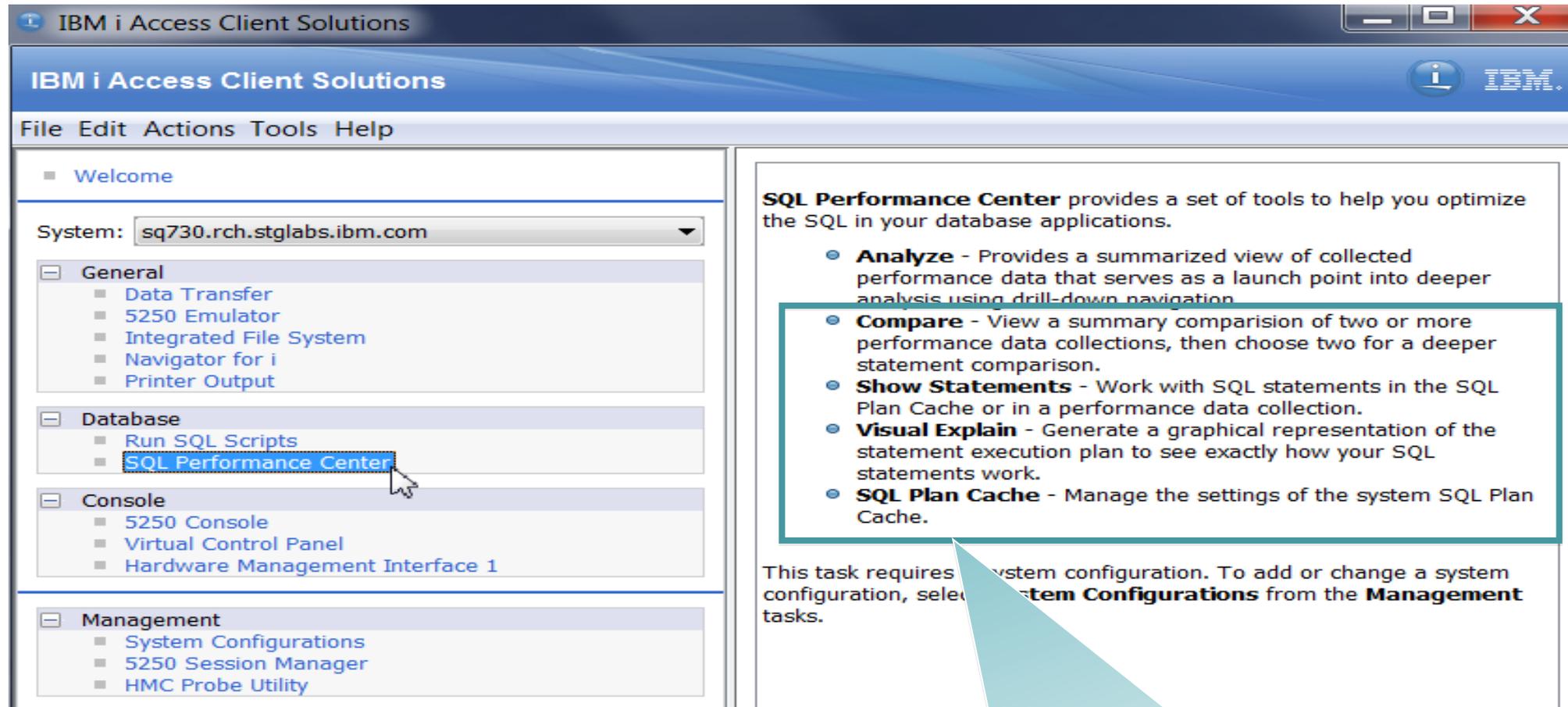
| Database Navigator Product Overview: | | | |
|---|--|---|--|
| Products | IBM i Access for Windows | IBM i Access Client Solutions | IBM i Navigator |
| Aliases | System i Navigator or IBM i Navigator | ACS | Navigator for i |
| Where does it run? | Windows PC Install | Anywhere Java can be used | Browser Served from IBM i 7.1 & 7.2 |
| Recent service level? | IBM i Access Windows Service Pack 7.1 – SI57907 | Version 1.1.5.0 | IBM HTTP SERVER FOR i PTF Group: 7.2 - SF99713 Level 12 7.1 - SF99368 Level 37 |
| Best of breed features | Run SQL Scripts Visual explain Many other | Run SQL Scripts SQL Performance Center | PDI Perspectives OmniFind administration Historical Collection Services Data |
| Webpage to watch | www-03.ibm.com/systems/power/software/i/access/windows_sp.html | http://www-03.ibm.com/systems/power/software/i/access/solutions_support.html | www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99713 www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99368 |
| Next (planned) Update | June, 2016 | July, 2016 è Version 1.1.6.0 | New PTF Group every 2-3 months |



iAccess for Windows 7.1 includes server-aware DB2 for i features such as RCAC, Temporal and more...

<http://www.ibm.com/support/docview.wss?uid=nas8N1019797>

August 2016: Database & ACS enhancement timeline



The screenshot shows the IBM i Access Client Solutions application window. The title bar reads "IBM i Access Client Solutions". Below the title bar is a menu bar with "File", "Edit", "Actions", "Tools", and "Help". The main content area is divided into a left-hand navigation pane and a right-hand main pane. The navigation pane has a "Welcome" section and a "System:" dropdown menu set to "sq730.rch.stglabs.ibm.com". Below this are three expandable sections: "General", "Database", and "Management". The "Database" section is expanded, and the "SQL Performance Center" item is highlighted with a mouse cursor. The right-hand pane displays the "SQL Performance Center" help text, which includes a list of features: Analyze, Compare, Show Statements, Visual Explain, and SQL Plan Cache. A light blue callout box with a pointer highlights the "Compare" feature description.

SQL Performance Center provides a set of tools to help you optimize the SQL in your database applications.

- **Analyze** - Provides a summarized view of collected performance data that serves as a launch point into deeper analysis using drill-down navigation.
- **Compare** - View a summary comparison of two or more performance data collections, then choose two for a deeper statement comparison.
- **Show Statements** - Work with SQL statements in the SQL Plan Cache or in a performance data collection.
- **Visual Explain** - Generate a graphical representation of the statement execution plan to see exactly how your SQL statements work.
- **SQL Plan Cache** - Manage the settings of the system SQL Plan Cache.

This task requires system configuration. To add or change a system configuration, select **System Configurations** from the **Management** tasks.

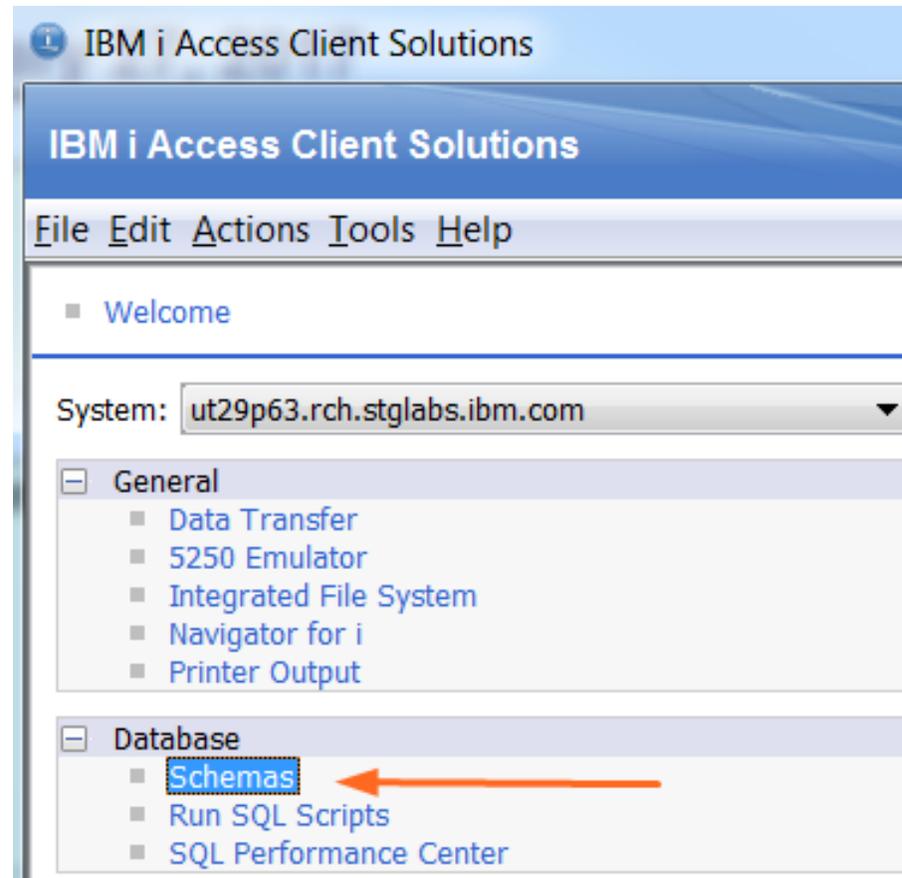
DBE's should be smiling

Database Tooling

ACS Version 1.1.7.0 (delivered in March, 2017)

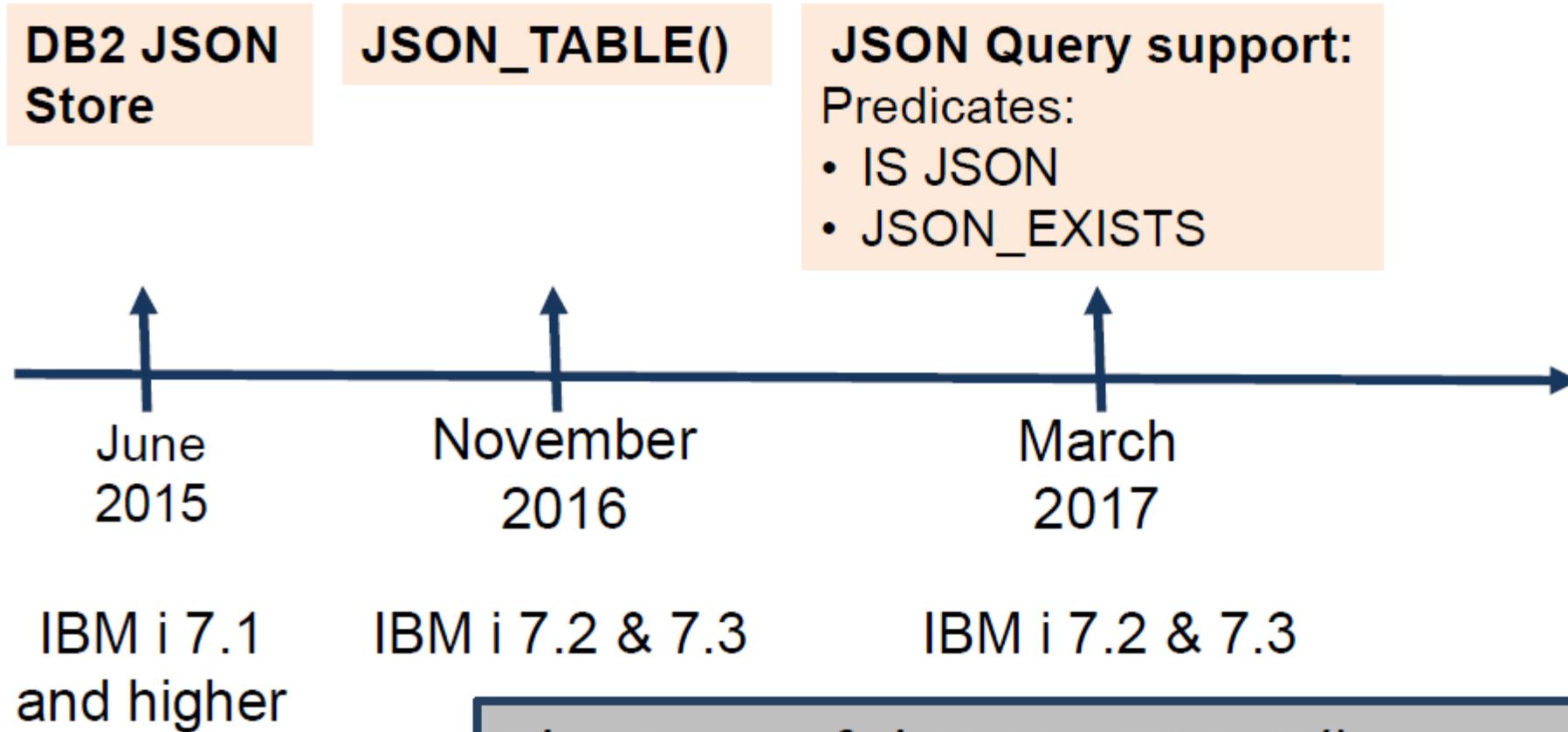
- **Run SQL Scripts**
 - CL Prompting
 - Enhanced SQL Highlighting
 - Show Object References
 - Upload Results
 - More “Insert from Examples”
- **Printer Output**
 - Multiple User Filter
- **IFS**
 - Filter

- **Schemas**



**IBM i 7.3 TR2,
7.2 TR6
&
DB2 for i**

Supporting New Data Types (only in SQL Tables) Example: JSON



The Powerful JSON_TABLE()
<https://www.ibm.com/developerworks/ibmi/library/i-json-table-trs/index.html>

JSON and the City of Chicago

← → ↻ 🏠 Secure https://data.cityofchicago.org/Transportation/Towed-Vehicles/ygr5-vcbg 🔍 ☆

Towed Vehicles

Transportation

View Data ↗ Download API Share ⋮

This dataset displays location for vehicles that have been towed and impounded in Chicago within the last 90 days. Illegally parked vehicles, abandoned vehicles and vehicles involved in illegal activities may be towed by the Chicago Police Department, the Department of Public Safety and Sanitation, the Department of Revenue, Aviation and the office of the City Clerk. When a tow request is issued, an inventory number is assigned by the Department of Street

About this Dataset

Updated Metadata

Access this Dataset via SODA API

The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data.

 API Docs  Developer Portal

API Endpoint

`https://data.cityofchicago.org/resource/rp42-fxjt.js` Copy

JSON and the City of Chicago

← → ↻ 🏠 Secure | <https://data.cityofchicago.org/resource/rp42-fxjt.json> 🔍 ☆ 📺 🗨

```
[{"color":"BLK","inventory_number":"965569","make":"ACUR","plate":"Y733803","state":"IL","style":"4D","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(312) 744-7550","towed_to_address":"400 E. Lower Wacker"},  
{  
"color":"SIL","inventory_number":"6884576","make":"BMW","plate":"Q915038","state":"IL","style":"LL","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(773) 265-7605","towed_to_address":"701 N. Sacramento"},  
,  
{"color":"BLK","inventory_number":"2820720","make":"BUIC","plate":"968XPC","state":"WI","style":"4D","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(773) 568-8495","towed_to_address":"10300 S. Doty"},  
,  
{"color":"GRY","inventory_number":"965570","make":"BUIC","plate":"N768966","state":"IL","style":"4D","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(312) 744-7550","towed_to_address":"400 E. Lower Wacker"},  
,  
{"color":"MAR","inventory_number":"2820738","make":"CADI","plate":"Q716098","state":"IL","style":"4D","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(773) 568-8495","towed_to_address":"10300 S. Doty"},  
,  
{"color":"RED","inventory_number":"6884578","make":"CHEV","plate":"Y865623","state":"IL","style":"2D","tow_date":"2017-04-25T00:00:00.000","tow_facility_phone":"(773) 265-7605","towed_to_address":"701 N. Sacramento"}]
```

JSON and the City of Chicago

- Once the feed is discovered, consumption is simple
- JSON_TABLE() requires a valid JSON object

```
CREATE OR REPLACE VARIABLE towed_vehicles
    CLOB(1G) CCSID 1208 ;

SET towed_vehicles = '{ "stuff" :' concat
    systools.HTTPGETCLOB(
    'https://data.cityofchicago.org/resource/rp42-fxjt.json',
    '')
    concat '}' ;
```

Most frequent Tow days

```
select TOW_DATE, DAYNAME(TOW_DATE) AS DAYNAME,  
       COUNT(*) AS TOW_COUNT  
from JSON_TABLE(towed_vehicles, 'lax $.stuff[*]' COLUMNS (  
TOW_DATE DATE      PATH '$."tow_date" '  
)) x  
group by TOW_DATE ORDER BY TOW_COUNT DESC;
```

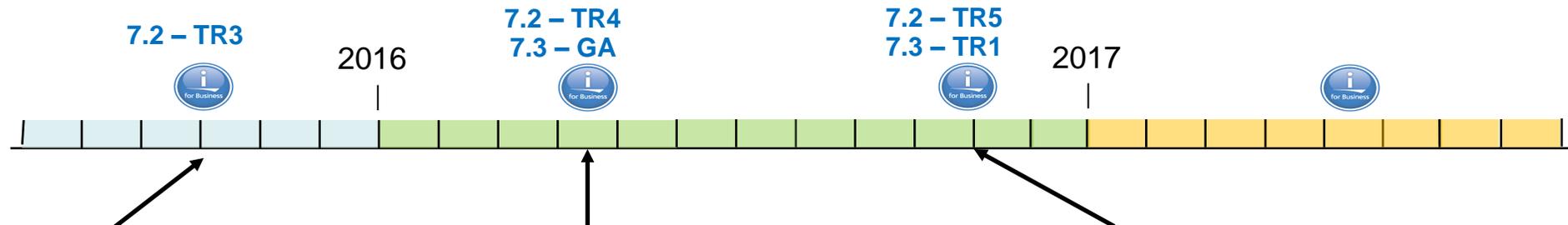
| <i>TOW_DATE</i> | <i>DAYNAME</i> | <i>TOW_COUNT</i> |
|-------------------|-----------------|------------------|
| <i>2017-04-23</i> | <i>Sunday</i> | <i>120</i> |
| <i>2017-04-15</i> | <i>Saturday</i> | <i>109</i> |
| <i>2017-04-21</i> | <i>Friday</i> | <i>104</i> |
| <i>2017-04-22</i> | <i>Saturday</i> | <i>100</i> |
| <i>2017-04-20</i> | <i>Thursday</i> | <i>97</i> |

Busiest tow lots

```
select TOW_TO_ADDRESS, COUNT(*) AS TOW_COUNT
  from JSON_TABLE(towed_vehicles, 'lax $.stuff[*]'
COLUMNS (
  TOW_TO_ADDRESS varchar(50) CCSID 1208
                                     PATH '$."towed_to_address"'
)) x
group by TOW_TO_ADDRESS ORDER BY TOW_COUNT DESC;
```

| <i>TOW_TO_ADDRESS</i> | <i>TOW_COUNT</i> |
|----------------------------|------------------|
| <i>701 N. Sacramento</i> | <i>522</i> |
| <i>10300 S. Doty</i> | <i>448</i> |
| <i>400 E. Lower Wacker</i> | <i>30</i> |

IBM i Services for SQL users



Services in 4Q/2015:

- MEMORY_POOL_INFO
- SYSTEM_STATUS_INFO
- LICENSE_INFO
- OBJECT_LOCK_INFO
- RECORD_LOCK_INFO
- OUTPUT_QUEUE_ENTRIES
- MEDIA_LIBRARY_INFO
- NETSTAT_INFO
- NETSTAT_JOB_INFO
- NETSTAT_INTERFACE_INFO
- NETSTAT_ROUTE_INFO
- GROUP_PTF_DETAILS

Services in 2Q/2016:

- OUTPUT_QUEUE_INFO
- ENVIRONMENT_VARIABLE_INFO
- SERVICES_INFO
- Services dependency management
- AUTHORITY_COLLECTION
- Enhanced:
 - NETSTAT services
 - SET_SERVER_SBS_ROUTING
 - System Limits
 - DISPLAY_JOURNAL()
 - OBJECT_STATISTICS

Services in 4Q/2016:

- **HISTORY_LOG_INFO**
- **JOB_INFO**
- **PARSE_STATEMENT**
- **Enhanced:**
 - **GET_JOB_INFO**
 - **GROUP_PTF_CURRENCY**
 - **SYSPARTITIONSTAT**
 - **DISPLAY_JOURNAL()**
 - **OBJECT_STATISTICS**
 - **Performance improvement for five services**

<http://ibm.biz/DB2foriServices>

IBM i Services – What happened in the last 2 TRs?

IBM® i Services

Security Services

QSYS2.AUTHORITY_COLLECTION – VIEW
QSYS2.AUTHORIZATION_LIST_INFO – VIEW
QSYS2.AUTHORIZATION_LIST_USER_INFO – VIEW
 QSYS2.DRDA_AUTHENTICATION_ENTRY_INFO – VIEW
 QSYS2.FUNCTION_INFO – VIEW
 QSYS2.FUNCTION_USAGE – VIEW
 QSYS2.GROUP_PROFILE_ENTRIES – VIEW
QSYS2.OBJECT_PRIVILEGES – VIEW
 QSYS2.SQL_CHECK_AUTHORITY – UDF
 QSYS2.USER_INFO – VIEW
 SYSPROC.SET_COLUMN_ATTRIBUTE – PROCEDURE

Communication Services

QSYS2.NETSTAT_INFO – VIEW
 QSYS2.NETSTAT_INTERFACE_INFO – VIEW
 QSYS2.NETSTAT_JOB_INFO – VIEW
 QSYS2.NETSTAT_ROUTE_INFO – VIEW
 QSYS2.SERVER_SBS_ROUTING – VIEW
 QSYS2.SET_SERVER_SBS_ROUTING – PROCEDURE
 QSYS2.TCPIP_INFO – VIEW
 SYSIBMADM.ENV_SYS_INFO – VIEW

Product Services

QSYS2.LICENSE_INFO – VIEW
SYSTOOLS.LICENSE_EXPIRATION_CHECK – PROCEDURE

Application Services

QSYS2.ENVIRONMENT_VARIABLE_INFO – VIEW
 QSYS2.QCMDXC – PROCEDURE
 QSYS2.SERVICES_INFO – TABLE
QSYS2.SET_PASE_SHELL_INFO – PROCEDURE

Storage Services

QSYS2.MEDIA_LIBRARY_INFO – VIEW
 QSYS2.SYSDISKSTAT – VIEW
 QSYS2.SYSTMPSTG – VIEW
QSYS2.USER_STORAGE – VIEW

Journal Services

QSYS2.DISPLAY_JOURNAL – UDTF
 QSYS2.JOURNAL_INFO – VIEW

Java Services

QSYS2.JVM_INFO – VIEW
 QSYS2.SET_JVM – PROCEDURE

Spool Services

QSYS2.OUTPUT_QUEUE_ENTRIES – VIEW
 QSYS2.OUTPUT_QUEUE_ENTRIES – UDTF
 QSYS2.OUTPUT_QUEUE_INFO – VIEW

Librarian Services

QSYS2.LIBRARY_LIST_INFO – VIEW
QSYS2.OBJECT_STATISTICS – UDTF

System Health Services

QSYS2.SYSLIMITS – VIEW
 QSYS2.SYSLIMTBL – TABLE

Message Handling Services

QSYS2.HISTORY_LOG_INFO – UDTF
 QSYS2.JOBLOG_INFO – UDTF
QSYS2.MESSAGE_QUEUE_INFO – VIEW
 QSYS2.REPLY_LIST_INFO – VIEW

PTF Services

QSYS2.GROUP_PTF_INFO – VIEW
 QSYS2.PTF_INFO – VIEW
SYSTOOLS.GROUP_PTF_CURRENCY – VIEW
 SYSTOOLS.GROUP_PTF_DETAILS – VIEW

Work Management Services

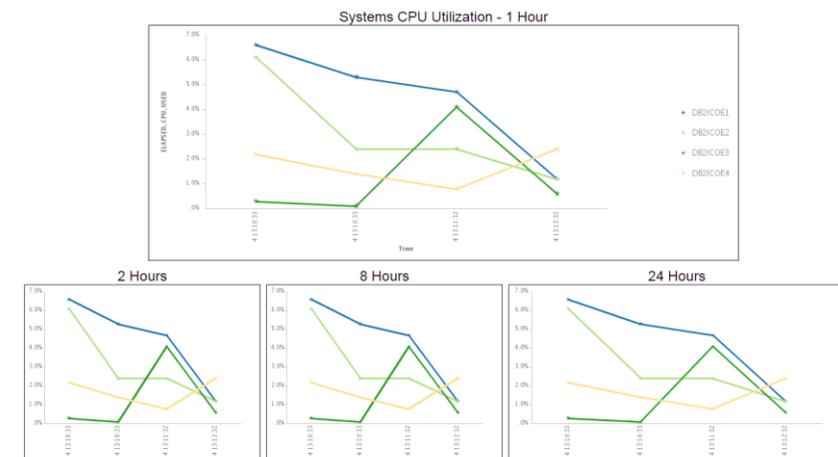
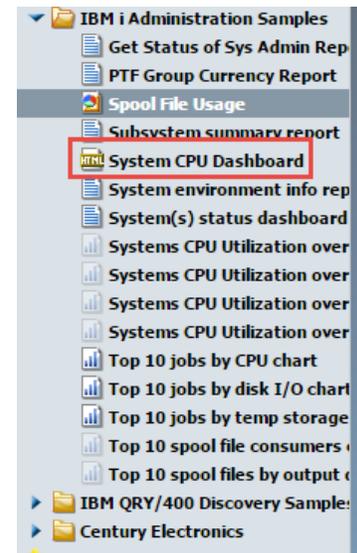
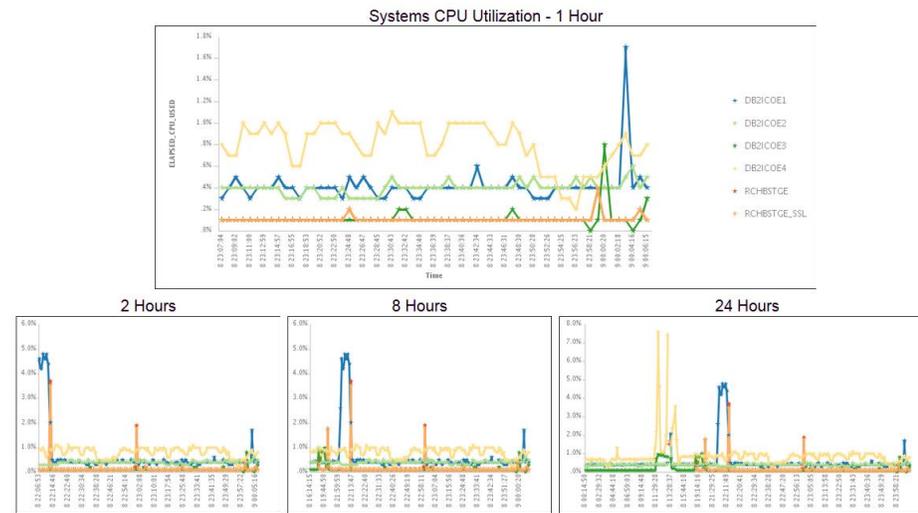
QSYS2.ACTIVE_JOB_INFO – UDTF
QSYS2.GET_JOB_INFO – UDTF
QSYS2.JOB_INFO – UDTF
 QSYS2.MEMORY_POOL – UDTF
 QSYS2.MEMORY_POOL_INFO – VIEW
QSYS2.OBJECT_LOCK_INFO – VIEW
 QSYS2.RECORD_LOCK_INFO – VIEW
 QSYS2.SCHEDULED_JOB_INFO – VIEW
 QSYS2.SYSTEM_STATUS – UDTF
 QSYS2.SYSTEM_STATUS_INFO – VIEW
 QSYS2.SYSTEM_VALUE_INFO – VIEW

DB2 Web Query and IBM i Services

- What are IBM i Services?
 - SQL Based solutions to common CL or API type commands
 - Delivered as SQL Views, User Defined Functions or Stored Procedures
 - All of which can be made available for use in DB2 Web Query reports
 - SQL Script examples provided in iNav and ACS Run SQL Scripts
 - Easier to get system information via SQL Based visualization tools like....DB2 Web Query
- Services are there today
 - Depending on TR and IBM i OS level
- DB2 Web Query can be leveraged to build reports, charts/graphs, dashboards or populate spreadsheets over the data returned from the various services
- Sample reports to be shipped in Q4 – stay tuned to my blog: [db2webqueryi@blogspot.com](http://db2webqueryi.blogspot.com)

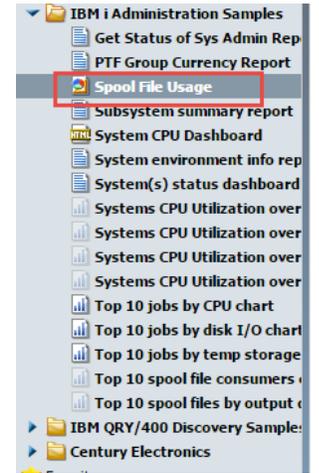
CPU Utilization Dashboard

- Supports one or multiple LPARs/Servers
 - System CPU Utilization
 - Can be easily extended to other system metrics
- Auto Refresh Enabled
 - Remote data collection requires remote system connections to be set up (RDB directory entries and SVRAUTE entries)
- Continues to collect and store historical data using a Stored Procedure once the report is initially run
 - Stops collecting if 'no one is watching' for an extended period of time

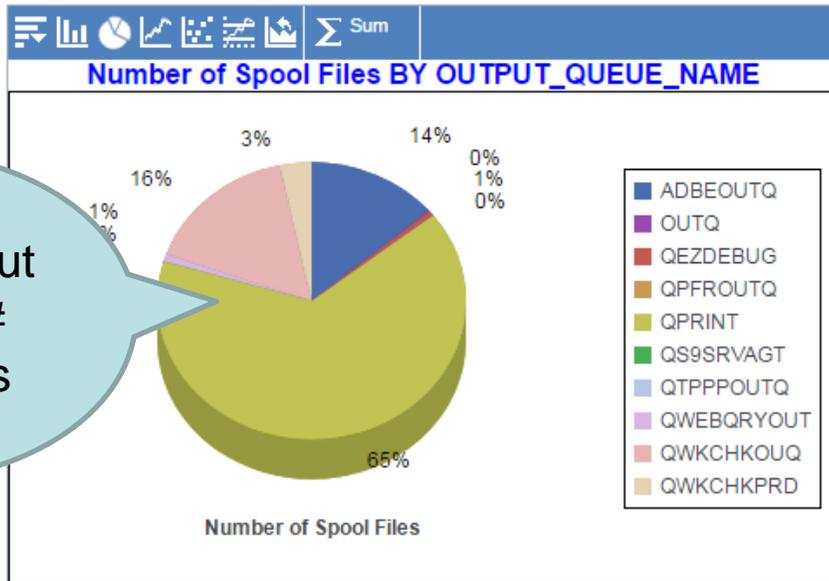


Monitoring Output Queues and Spool Files

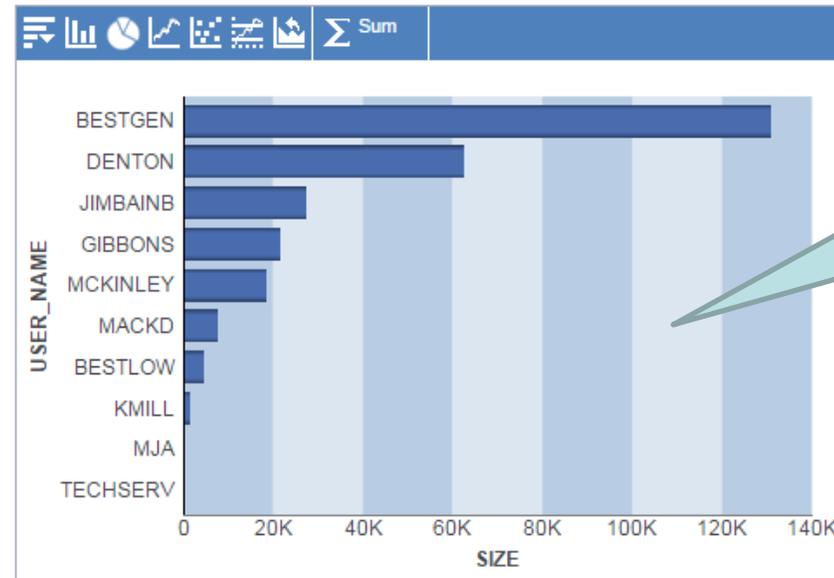
- Are you cleaning up your spool files?



Output Queue Top 10 at 13.36.43 on May 4, 2017



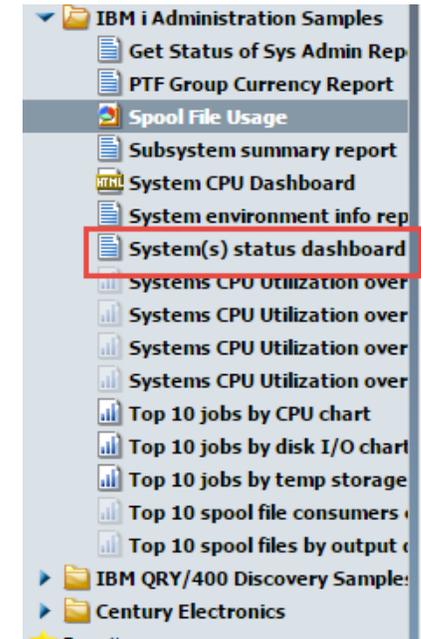
Top 10 output queues by # of spool files



Top 10 users by # of spool files

System Status

- Various metrics across all LPARs/Servers
 - Not set up as auto refresh but could be easily enough
 - Traffic lighting to highlight potential problems
 - You could easily change the report to include your own thresholds to monitor



System(s) Status

| Host Name | Total Jobs in System | Active Jobs in System | Interactive Jobs in System | Active Threads in System | Elapsed Time | Elapsed CPU Used | Current CPU Capacity | Average CPU Utilization | Minimum CPU Utilization | Maximum CPU Utilization | System ASP Storage | Total Auxiliary Storage | System ASP Used | Current Temporary Storage | Maximum Temporary Storage |
|-----------|----------------------|-----------------------|----------------------------|--------------------------|--------------|------------------|----------------------|-------------------------|-------------------------|-------------------------|--------------------|-------------------------|-----------------|---------------------------|---------------------------|
| DB2ICOE1 | 398 | 263 | .00 | 1188 | 1 | 1.30 | 2.00 | .18 | .00 | 99.95 | 1488922 | 1488922 | 61.04 | 7529 | 7585 |
| DB2ICOE2 | 6441 | 244 | .00 | 1112 | 1 | .20 | 2.00 | .21 | .00 | 99.76 | 954437 | 954437 | 50.21 | 7778 | 7846 |
| DB2ICOE3 | 8934 | 282 | .00 | 1218 | 1 | .70 | 4.00 | .11 | .00 | .28 | 6242936 | 6242936 | 27.83 | 8215 | 8260 |
| DB2ICOE4 | 4869 | 242 | .00 | 925 | 1 | 2.80 | 2.00 | .11 | .00 | .14 | 6242936 | 6242936 | 15.73 | 6976 | 7833 |

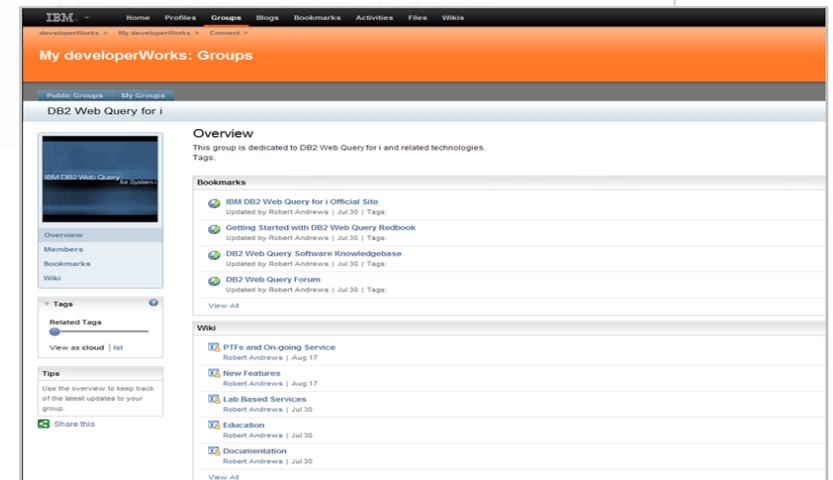
Run on May 4, 2017 at 13.46.47

Available as Part of DB2 Web Query EZ-Install

- Earlier this year we introduced a simple way of installing and getting immediate value out of DB2 Web Query
 - EZ-Install Package
 - Includes Sample reports/dashboards that a Systems Admin might be interested in
 - DB2 Web Query audit reports/dashboards
 - Automated setup so you can immediately run with the over 14 chapters of self guided tutorials
 - And we also include the completed tutorial reports so you have over 50 sample reports/dashboards
 - Query/400 Discovery Tool
 - Must be at 7.1 or above
- To Request, send email to QU2@us.ibm.com
 - Include name, company name, and serial number where you plan to install

To Learn More

- DB2 Web Query for i Website
 - [Ibm.biz/db2webqueryi](http://ibm.biz/db2webqueryi)
- DB2 Web Query for i Wiki
 - [Ibm.co/db2wqwiki](http://ibm.co/db2wqwiki)
- DB2 Web Query Getting Started Enablement:
 - <https://ibm.biz/db2wqconsulting>
- Demonstrations:
 - Wizard Analytics: <https://ibm.biz/DB2WQWizards>
 - End User Demos: <https://ibm.biz/db2wqreportingdemos>
 - Getting Started Videos: <https://ibm.biz/db2wqgettingstarteddemos>
- Follow DB2 Web Query guy Doug Mack on twitter at @mckdrmoly or check out his blog at <http://db2webqueryi.blogspot.com/> for all the latest

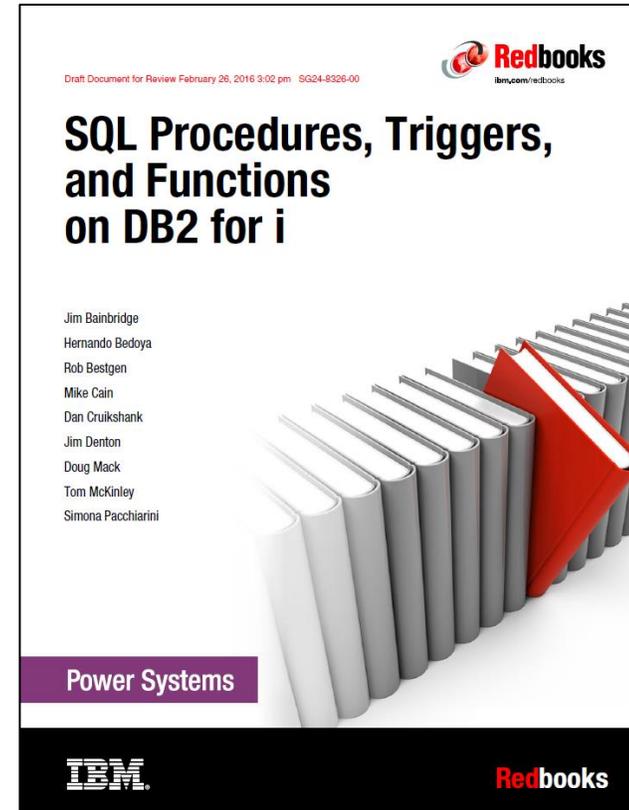


DB2 for i New/Updated Redbooks

DB2 for i – SQL Programming Resources

**Essential resource for SQL & DB2
for i database application
development**

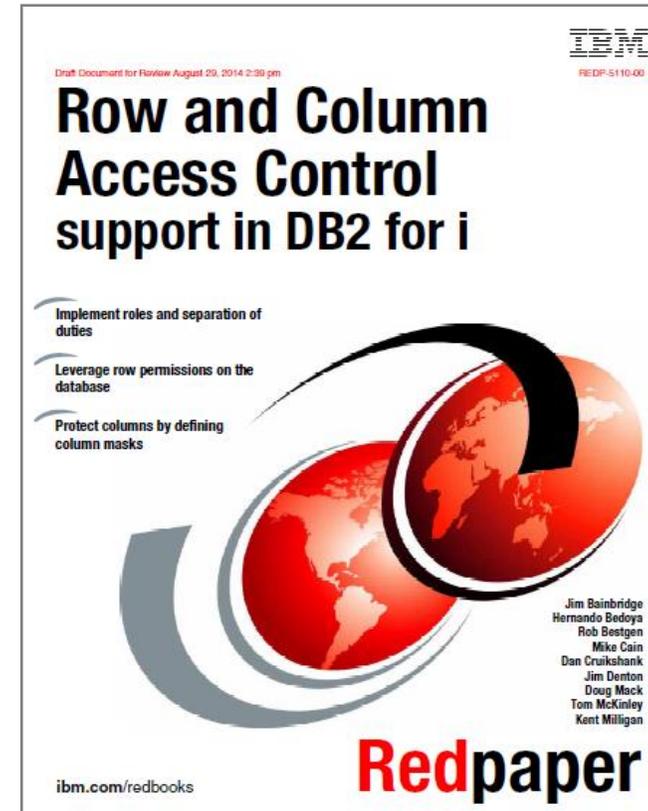
Draft published: March 2016



www.redbooks.ibm.com/redpieces/abstracts/sg248326.html

DB2 for i – RCAC Resource

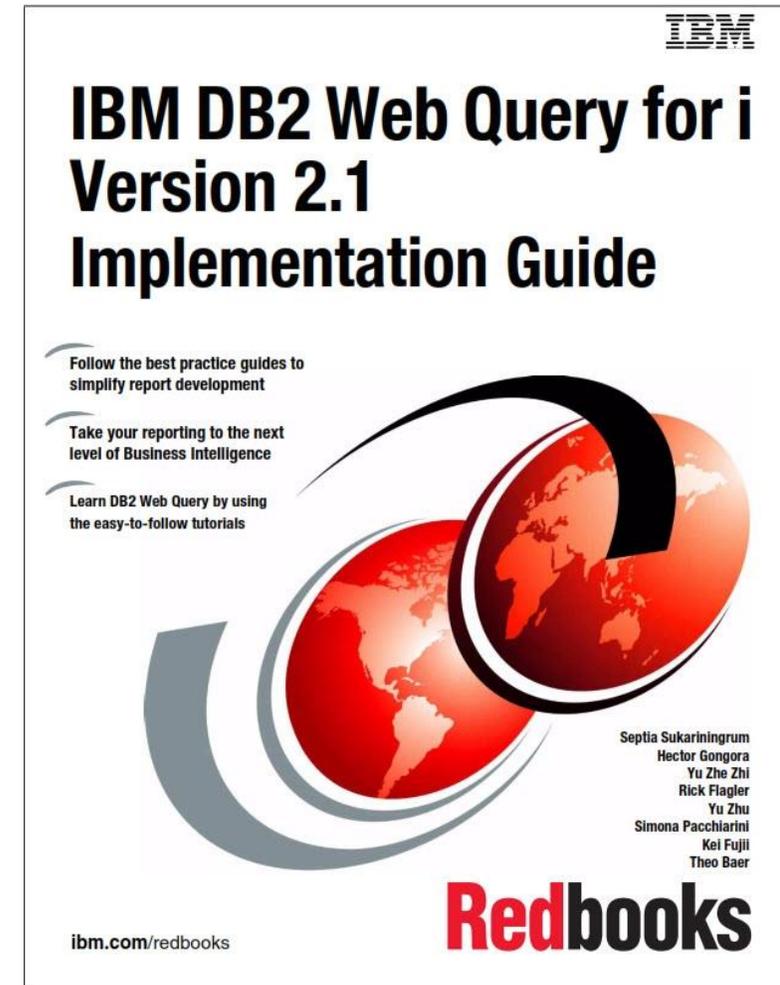
Essential resource for understanding how Row Permissions and Column Masks can be used to limit access to business critical data



www.redbooks.ibm.com/redpieces/abstracts/redp5110.html

COMING SOON: Two New Redbooks for DB2 Web Query 2.2

- **Current Version 2.1 Redbook contains over 14 chapters of Tutorials**
 - All will be updated
 - New ones are being added
- **New “Nuts and Bolts” Redbook**
 - Many more topics previously not covered or updated to cover new function





ithankyou

www.ibm.com/developerworks/ibmi/techupdates/db2