DB2 for IBM i Database Modernization: Object Creation and Access

Tom Mckinley (Mac2@us.ibm.com)
Lab Services
IBM Rochester

© Copyright IBM Corporation, 2009. All Rights Reserved. This publication may refer to products that are not currently available in your country. IBM makes no commitment to make available any products referred to herein.
Agenda

• Why modernize with SQL & DB2?
• Approaches & Options
• Modernizing Database Definitions
• Modernizing Data Access
• Next Steps
Why SQL?

• Portability of code & skills

• Strategic database interface for industry & i5/OS
  – Faster performance delivered by SQE only available to SQL-based interfaces
  – SQL required for certain functions & middleware
    • J2EE architecture based on SQL interfaces
    • Data types: BLOB, CLOB, Datalink, ...
    • Auto-Incrementing Constructs: Sequence & Identity column attribute
    • Column-level Triggers
    • Encryption & Decryption functions
    • Encoded Vector Indices
    • ...

• Enables better positioning of System i as a Database Server

• SQL as a programming language can reduce total lines of code

• DB2 SMP - parallel database processing

The New Power Equation
Approaches & Options

ODBC / JDBC / ADO / DRDA / XDA

Network

Host Server

CLI / JDBC

Static
Compiled embedded statements

Dynamic
Prepare every time

Extended Dynamic
Prepare once and then reference

Native (Record I/O)

SQL

Optimizer

DB2
(Data Storage & Management)
Approaches & Options

SQL-created objects

DDS-created objects

SQL Programs

Native* Programs

Considerations: Multi-member & multi-format files

*Restrictions: EVIs, LOB columns, UDTs, Datalinks, etc
Modernizing Definitions & Objects

• Modeling

• Terminology

• Moving from DDS to SQL DDL

• SQL object management

• Embedding business logic into database definitions
Modernizing Definitions & Objects

Data modeling

- “Master data” concept
  - Services created to retrieve data – what if multiple copies exist?

- Database normalization
  - Define a separate table for each related set of values
  - Define the primary key (surrogate or natural)
  - Eliminate redundant data
  - Fifth normal form (5NF) recommended, 3NF at minimum
  - Establish RI constraints

- Also consider
  - Globalization
Modernizing Definitions & Objects

Data Modeling IBM Rational Data Architect (Version 7)

- Enterprise data modeling and management
  - Compare & synchronize
  - Forward & reverse engineering
  - Logical file support – Fixpack 003
  - Model analyzer for enterprise standard conformance
- Database development – SQL Stored Procedures and Function
- Trial Download: ibm.com/software/data/integration/rda/

The New Power Equation
## Modernizing Database Objects

### Terminology

<table>
<thead>
<tr>
<th>SQL</th>
<th>i5/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>schema/collection</td>
<td>library</td>
</tr>
<tr>
<td>table</td>
<td>physical file</td>
</tr>
<tr>
<td>view</td>
<td>logical file</td>
</tr>
<tr>
<td>index</td>
<td>keyed logical file</td>
</tr>
<tr>
<td>row</td>
<td>record</td>
</tr>
<tr>
<td>column</td>
<td>field</td>
</tr>
<tr>
<td>log</td>
<td>journal</td>
</tr>
</tbody>
</table>
Tables vs PFs

- SQL Tables compared with Physical Files
  - **Advantages**
    - More data types
    - Constraint definitions can be included in object source
    - Faster reads
    - Longer, more descriptive column names
    - Data Modeling Tool support
    - DB2 attempts to automatically journal tables
  - **Disadvantages**
    - Slower writes
    - No DDM, BUT SQL can utilize DRDA connections
    - Multi-member files
      - SQL ALIAS provides solution:
        CREATE ALIAS JanSales FOR SALES (JANUARY )
Modernizing Database Definitions & Objects

Indexes vs LFs

• SQL Indexes compared with Keyed Logical Files
  – Advantages
    • Encoded Vector Index Structure
    • 64K Logical Page Size (since V4R2) – more aggressive input
  – Disadvantages
    • 8K Logical Page Size – less aggressive input
    • No support for Select/Omit filtering or join logical files
      – V6R1 provides new syntax to allow the creation of SQL indexes with
        selection criteria and derivations
V6R1 – SQL Derived Index Example

**DDS for Existing Select/Omit LF:**

```
A          R ITEM_FACT
A          ORDERKEY
A          SHIPMODE
A          PARTKEY
A          K ORDERKEY
A          K SHIPMODE
A          S SHIPMODE
A          K SHIPMODE
```

- **Equivalent SQL syntax:**

```
CREATE INDEX sql_selectomit
ON item_fact ( orderkey, shipmode )
WHERE shipmode = 'MAIL'
RCDFMT item_fact ADD partkey
```
Modernizing Database Definitions & Objects

Views vs LFs

- SQL Views compared with Logical Files
  - Advantages
    - More flexibility in terms of selecting & processing data
      - CASE expressions & Date/Time functions
      - Grouping & more advanced Join processing
    - Can be used as logical files to enhance native functionality
  - Disadvantages
    - Views cannot be keyed/ordered
      - Does that mean SQL Views have slower performance?

- NO - assuming you have the right set of indexes/statistics in place for the query optimizer to use
- View is used by SQL just to transform data, query optimizer's job to find the best method to speed up selection or sorting
- Fastest method may not be a keyed access method
Modernizing Database Definitions & Objects

DDS to SQL Conversion

- System i Navigator Generate SQL Task (QSQGNDDDL API)
  - Useful in converting object definitions from DDS to SQL
  - Supports physical & logical files
    - Not all DDS features can be converted, tool will convert as much as possible and generate warnings for unconvertible options (e.g., EDTCDE)
    - Logical files converted to SQL Views
    - SQL Field Reference File support not used
  - Can convert a single object or a group of objects
  - Output can be edited & saved directly into source file members
Modernizing Database Definitions & Objects

SQL Object Compatibility

• SQL Column & Object names have maximum lengths of 30 & 128, but many i5/OS utilities, commands and interfaces only support a 10-character length. How does that work?!?!
  – System automatically generates a short 10 character name
    • First 5 chars with unique 5 digit number
      CUSTOMER_MASTER >> CUST00001
  • Might be different each time a specific table is created, depending on creation order and what other objects share the same 5 character prefix
  • Use i5/OS SQL syntax to specify your own short name
    – RENAME TABLE (tables & views) & RENAME INDEX
    – FOR COLUMN clause for columns
    – SPECIFIC clause for procedures, functions
Modernizing Database Definitions & Objects

SQL Object Compatibility

• Short & Long Name Co-existence Example
  – Specify the short name at creation:

    CREATE TABLE dbtest/cusmst
    (customer_name FOR COLUMN cusnam CHAR(20),
     customer_city FOR COLUMN cuscty CHAR(40))

  – Specify a long name for existing short-name:

    RENAME TABLE dbtest/cusmst TO customer_master
    FOR SYSTEM NAME cusmst

• If long name specified on SQL Table definition, can also add/control the short name after table created:

    RENAME TABLE dbtest/customer_master TO SYSTEM NAME cusmst
Modernizing Database Definitions & Objects

SQL Object Compatibility

- RENAME statement can also be used to control the format & file name used by native programs
  - Specify the format name (cmfmt) when creating the table
    
    ```sql
    CREATE TABLE dbtest/cmfmt
    (customer_name FOR COLUMN cusnam CHAR(20),
     customer_city FOR COLUMN cuscty CHAR(40))
    ```

  - Use RENAME to specify the short file name (cusmst) for table:
    ```sql
    RENAME TABLE dbtest/cmfmt TO cusmst
    ```

  - Use RENAME to specify the long name for SQL interfaces
    ```sql
    RENAME TABLE dbtest/cusmst TO customer_master
    FOR SYSTEM NAME cusmst
    ```

- V5R4 simplifies control of the record format name with new SQL keyword...
  ```sql
  CREATE TABLE dbtest/customer_master
  (customer_name FOR COLUMN cusnam CHAR(20),
   customer_city FOR COLUMN cuscty CHAR(40))  RCDFMT cmfmt
  ```

The New Power Equation
Modernizing Database Definitions & Objects

SQL Object Compatibility

• What happens to my existing applications if my objects are now created with SQL instead of DDS?

IT DEPENDS on your approach!
Modernizing Database Definitions & Objects

SQL Object Compatibility

• Recommend Methodology (documented in IBM Redbook)
  – Convert Physical file source (PF1) into SQL Table (TAB1)
  – Delete physical file PF1
  – Convert the PF1 DDS source into a Logical file named PF1 that references TAB1
  – Change the source for all logical files over PF1 to share format of PF1 and reference TAB1
    R PF1 FILER PFILE(TAB1) FORMAT(PF1)

• XCase for System i new tool available to automate & manage this conversion methodology
  – Ensure the right steps are done in proper order
  – Data modeling tool also available

The New Power Equation
Modernizing Database Definitions & Objects

SQL Object Management

• SQL Source Management best practices:
  – Store SQL source in source physical file members just like DDS and “execute” with the RUNSQLSTM instead of CRTPF/CRTL
    • Store SQL scripts in PC or IFS files for non-i5/OS change management tools
    • Generate SQL can be used to retrieve misplaced SQL source from System Catalogs (SYSIBM & QSYS2)
    • V6R1 RUNSQLSTM supports IFS files and wider margins & CL commands
  – SQL Table definitions can use Field Reference File
    CREATE TABLE customer AS
      (SELECT id cust_id, Iname cust_lastname, fname cust_firstname, 
       city cust_city FROM RefFile)
    WITH NO DATA

• May need to adjust process for moving from development to production
  – Best practice is to re-execute SQL creation script
  – Save/Restore process for SQL databases documented at:
    ibm.com/developerworks/db2/library/techarticle/0305milligan/0305milligan.html
Modernizing Definitions & Objects

SQL & Non-relational data

• User-Defined Table Functions
  – Allows non-relational & legacy data to be virtualized as an SQL table

  SELECT * FROM TABLE(myudtf('Part XYZ'))

  – Both SQL & External Table Functions supported
    • External UDTFs can be easily written to access multi-format files, S/36 files, and stream files
    • Table functions can only be invoked from SQL-based interfaces

• Datalinks
  – URL-based data type to provide linkage to related objects in IFS
  – Can establish RI relationship between table row & IFS object

• LOBs
  – Allows you to keep non-relational data along with all the other business data

The New Power Equation
Moving Business Logic into DB2 - Automatic Key Generation

• Identity Column Attribute
  – Attribute that can be added to any numeric columns to have DB2 generate next value
  – Not guaranteed to be unique, primary key or unique index must be defined
  – Only available for SQL tables, BUT identity column value generated for both SQL and non-SQL interfaces (RPG, etc) that are adding new rows

  CREATE TABLE emp( empno INTEGER GENERATED ALWAYS AS IDENTITY
  (START WITH 10 , INCREMENT BY 10),
  name CHAR(30), dept# CHAR(4))

  INSERT INTO employee(name,dept) VALUES('MIKE','503A') or…
  INSERT INTO employee VALUES(DEFAULT,'MIKE', '503A')

• Sequence Object
  – Separate object that can be shared across multiple tables
  – Generated value to be part of non-numeric keys

  CREATE SEQUENCE order_seq START WITH 10 INCREMENT BY 10

  INSERT INTO orders(ordnum,custnum) VALUES( NEXT VALUE FOR order_seq, 123 )
Modernizing Definitions & Objects

Moving Business Logic into DB2 - Constraints

• Database Constraints Benefits
  – Easier code reuse & better modularity
  – Improved data integrity
  – Improved query performance - SQE query optimizer is constraint aware

• Constraint Types
  – Primary & Unique Key
  – Referential Integrity Constraints
    • Enforce Parent/Child & Master/Detail relationships
  – Check Constraints
    • Ensure that a column is only assigned legal values

CREATE TABLE orders(
    ordnum INTEGER PRIMARY KEY,
    ordqty INTEGER CHECK(ordqty>0 AND ordqty<999),
    ordamt DECIMAL(7,2),
    part_id CHAR(4),
    CONSTRAINT ordpart FOREIGN KEY(part_id) REFERENCES parts(PartID)
        ON DELETE RESTRICT  ON UPDATE RESTRICT )
### Modernizing Definitions & Objects

**Moving Business Logic into DB2 - Triggers**

- Triggers allow you initiate business policies & processes whenever new data comes in or existing data is changed
  - DB2 responsible for always invoking the trigger program
  - Execution independent of the user interface
  - Transform data before it gets into DB2

- **DB2 for i Trigger Support**
  - Before & After: Insert, Update, & Delete events (up to 300 triggers)
  - SQL & External(ADDPFTRG) Triggers
    - Column-level & Statement-level triggers only available with SQL Triggers

```sql
CREATE TRIGGER audit_salary
AFTER UPDATE ON employee(salary)
REFERENCING NEW AS n
REFERENCING OLD AS o
FOR EACH ROW
WHEN (n.salary - o.salary >= 5000)
INSERT INTO audit
VALUES(n.empno, n.deptno, n.salary, current timestamp)
```

**The New Power Equation**
Modernizing Data Access

• Programming Interfaces

• Native I/O to SQL Comparison

• Using SQL to Reuse & Repurpose Existing Code

• DB2 & New Application Models

• Tools
Modernizing Data Access

Programming Interfaces

<table>
<thead>
<tr>
<th>Static SQL</th>
<th>Dynamic SQL</th>
<th>Extended Dynamic SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Static</td>
<td>Embedded Dynamic</td>
<td>QSQPRCED</td>
</tr>
<tr>
<td></td>
<td>JDBC, SQLJ</td>
<td>Toolbox JDBC &amp; iSeries Access</td>
</tr>
<tr>
<td></td>
<td>OLE DB, .NET</td>
<td>ODBC</td>
</tr>
<tr>
<td></td>
<td>CLI, ODBC</td>
<td>XDA API set</td>
</tr>
<tr>
<td></td>
<td>PHP ibm_db2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUNSQLSTM</td>
<td></td>
</tr>
</tbody>
</table>

**DB2 SQL Development Kit only required if embedded SQL (& STRSQL) is going to be used**

*The New Power Equation*
Modernizing Data Access

Native I/O to SQL Example

... 
C/EXEC SQL 
C+ DECLARE sql_jn CURSOR FOR SELECT 
C+   t.year,t.month,i.orderdt,c.country,c.cust 
C+   p.part,s.supplier,i.quantity,i.revenue 
C+ FROM item_fact i 
C+ INNER JOIN part_dim p ON (i.partid=p.partid) 
C+ INNER JOIN time_dim t ON (i.orderdt=t.datekey) 
C+ INNER JOIN cust_dim c ON (i.custid=c.custid) 
C+ INNER JOIN supp_dim s ON (i.suppid=s.suppid) 
C+ WHERE year=1998 AND month=6 
C/END-EXEC

C/EXEC SQL 
C+ OPEN sql_jn 
C/END-EXEC

C/EXEC SQL 
C+ FETCH NEXT FROM sql_jn FOR :RowsReq ROWS 
C+ INTO :result_set 
C/END-EXEC

C If SQLCOD = 0 and 
C   SQLER5 = 100 and 
C   SQLER3 > 0 
C Eval RowsRd = SQLER3

...
... 
C/EXEC SQL
C+ DECLARE sql_jn CURSOR FOR
C+ SELECT * FROM JoinView
C+ WHERE year=1998 AND month=6
C/END-EXEC

C/EXEC SQL
C+ OPEN sql_jn
C/END-EXEC

C/EXEC SQL
C+ FETCH NEXT FROM sql_jn FOR
C+ :RowsReq ROWS INTO :result_set
C/END-EXEC

C If SQLCOD = 0 and
C SQLER5 = 100 and
C SQLER3 > 0
C Eval RowsRd = SQLER3

C SearchKey KList
C Kfld SearchYear
C Kfld SearchMonth

C SearchKey Setll NTVJOIN002
C If %FOUND
C DO RowsReq Times
C Times Occur Result_Set
C READ NTVJOIN002
C If %EOF
C Leave
C Endif
C Eval RowsRd = RowsRd + 1
C ENDDO
C Endif

Native I/O to SQL Example - Joined LFs & Views
Modernizing Data Access

Native I/O to SQL Example - Performance Comparison

Note: Tests run on Model 720 w/1600 CPW & 2 GB Memory - your performance results may vary

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native File Join</td>
<td>0.002512</td>
<td>0.260248</td>
<td>2.219504</td>
<td>23.228176</td>
</tr>
<tr>
<td>Native JoinLF</td>
<td>0.002304</td>
<td>0.362128</td>
<td>2.544608</td>
<td>21.366480</td>
</tr>
<tr>
<td>Native JoinLF w</td>
<td>0.002400</td>
<td>2.144288</td>
<td>2.125032</td>
<td>19.311464</td>
</tr>
<tr>
<td>SQL - No IOA</td>
<td>0.145160</td>
<td>0.489136</td>
<td>3.166704</td>
<td>20.452984</td>
</tr>
<tr>
<td>SQL IOA</td>
<td>0.251168</td>
<td>0.267208</td>
<td>0.417800</td>
<td>1.898800</td>
</tr>
<tr>
<td>SQL SQE IOA</td>
<td>0.013536</td>
<td>0.019320</td>
<td>0.250160</td>
<td>1.576536</td>
</tr>
</tbody>
</table>
Modernizing Data Access

Native to SQL Considerations

• ORDER BY clause is the only way to guarantee the sequencing of results when using SQL - no clause, means ordering by chance
• SQL Precompilers do not support all the latest features (free format SQL supported added in V5R4) - still missing from RPG Precompiler in V6R1:
  – Support for properly scoping a local variable in a subprocedure as a host variable in an SQL statement (ie, multi-pass) THIS WAS FIXED IN V6R1
  – Support for qualified names with more than one level of qualification
• Consider impact of SQL isolation level & journaling on native applications
• Critical Performance Success Factors
  – Sound Indexing & Statistics Strategy (ibm.com/servers/enable/site/education/ibo/record.html?indxng)
  – Reusable Open Data Paths (ODPs)
    • Prepare Once, Execute Many
    • Connection Pooling
    • Keep Connections & Jobs active as long as possible
    • Reference: ibm.com/servers/enable/site/education/abstracts/db2sql_abs.html
  – Blocked Fetches & Inserts
  – AVOID IMPLEMENTING THE RECORD AT A TIME PROCESSING IN SQL

The New Power Equation
Modernizing Data Access

Using SQL to Reuse & Repurpose Existing Code

- Stored Procedures, Functions, & Triggers provide vehicle for improving and changing the architecture of your solution
  - Improved modularity by allowing same code to be used by multiple interfaces & applications
  - Better partitioning of logic (eg, separation of presentation & database logic)
  - Easy transition to multi-tier architectures since many interfaces exist for remote invocations

- DB2 for i support provides maximum flexibility by supporting both SQL & External types
  - External support allows reuse of existing i5/OS application code & skills
  - SQL Procedural Language (PSM) enables better portability of logic (& programming skills) to/from other platforms
  - Data security can be enhanced/maintained with i5/OS Adopted Authority
Modernizing Data Access

Using SQL to Reuse & Repurpose Existing Code: User-Defined Functions

- UDFs allow the database to invoke user-written functions during the processing of an SQL statement
  - Allows you to customize SQL to meet your business requirements
  - Example:

    ```sql
    CREATE FUNCTION Euro(EuroAmt DECIMAL(11,2))
    RETURNS DECIMAL(11,2)
    LANGUAGE SQL
    BEGIN
      DECLARE rate DECIMAL(9,5);

      SELECT conversion_rate INTO rate FROM ratetable WHERE ...;
      RETURN rate*EuroAmt;
    END
    
    SELECT item_name, Euro(item_name) FROM parts...
    ```
Modernizing Data Access

Using SQL to Modernize Reporting

- **Query/400 → DB2 Web Query**
  - Browser interface
  - Multiple output options:
    - Excel, HTML, PDF, etc
  - Better performance with SQL-based reporting
  - Existing Query/400 reports can be imported
    - Web Query output capabilities can be applied to Query/400 results
    - Web Query input parameters can be added

- **OPNQRYF → SQL**
  - Retain “set at a time processing” but provide data using SQL
  - White paper on conversion considerations:
Modernizing Data Access

DB2 & New Application Models - XML & Text Mining

• DB2 Extenders provide low-level plumbing to allow you to concentrate on the business logic
• DB2 & XML Integration with DB2 XML Extender
  – Allows an XML document to be stored & retrieved from a column
  – Enables XML document to be generated dynamically from existing DB2 data
  – Provides ability to decompose an XML document & generate new rows in your database
  – Redbooks: The Ins and Outs of XML and DB2 for i5/OS
• Text Mining with DB2 Text Extender & IBM OmniFind – V6R1
  – High-speed, sophisticated searches for any character columns
    • Fuzzy searches
    • Search on tenses of word
    • Customize search to words in same sentence or paragraph
  – Can also search text documents stored in IFS
• More Extender details at:
  ibm.com/servers/enable/site/education/ibo/record.html?db2udbext
Modernizing Data Access

“SQL” Development Tools

• WebSphere Development Studio Client & Rational Developer for System i
  – DB2 Web Service Support
  – XML Extender Aids
  – Enhanced SQL Integration in 7.0 and 7.1 Clients

• Microsoft Visual Studio .NET Integration with V6R1 ADO.NET Provider

• DB2 Developer Workbench (DB2 Data Studio)
  – Stored Procedures
  – User-Defined Functions
  – SOA Wizards
  – Java pureQuery runtimes

• Toolbox Graphical Debugger for ILE & SQL-source level debug
  (ibm.com/servers/enable/site/education/abstracts/sqldebug_abs.html)

• XCase for System i – Support for DDS to SQL Migration & Data Modeling
“SQL” Development Tool Example

- SQL syntax highlighting for both Free and Fixed Format ILE RPG
  - Websphere Development Studio Client 7.0 & Rational Developer for System i 7.1

```
/free
  exec sql create procedure median_result_set
  language sql dynamic result sets 1
  BEGIN
    case v_workdept when 'B01'
      -- comments
      then update department set deptname = 'DATA ACCESS 2';
    else update department set deptname = 'DATA_ACCESS_3';
    end case;
  end;
/end-free

000100
000200
000300
000400
000500
000600
000700
000800
000900
001000
001100
001200
001300
001400
001500
001600
001700
001800
001900
002000
002100
002200
002300
002400
002500
002600

000100
000200
000300
000400
000500
000600
000700
000800
000900
001000
001100
001200
001300
001400
001500
001600
001700
001800
001900
002000
002100
002200
002300
002400
002500
002600
```

The New Power Equation
“SQL” Development Tool Example

- Customizable statement templates for ILE RPG Free format to accelerate SQL coding

CREATE TABLE ${CURSOR}table1 (  
col1 CHAR(3) NOT NULL,  
col2 INTEGER,  
col3 VARCHAR(28) NOT NULL,  
PRIMARY KEY (col1) );
Modernizing Data Access

“SQL” Development Tools

• IBM DB2 Web Query – ibm.com/systemi/db2/webquery
  – Redbook: Getting Started with DB2 Web Query for System i (SG24-7214)
• System i Navigator (iSeries Navigator)
  – Editors for procedure, functions, triggers
  – SQL statement wizard for INSERT, SELECT, UPDATE, DELETE
  – Downloadable Tutorials at:
    ibm.com/servers/enable/site/education/ibo/view.html?oc#db2
  – OnDemand Performance Center
    • Visual Explain
    • SQL Performance Monitors
    • SQL Plan Cache
    • System-Wide Index Advisor

• DB2 SMP - licensed feature (IBM i Option 26)
• Websphere Federation Server for non-DB2 data access
  ibm.com/servers/enable/site/education/ibo/record.html?hetdata
Next Steps

• EDUCATION
  – Modernizing iSeries Application Data Access Redbooks document
    www.redbooks.ibm.com/abstracts/sg246393.html?Open
  – Case Study: Modernizing a DB2 for iSeries Application white paper
    ibm.com/servers/enable/site/education/abstracts/9e5a_abs.html
  – DB2 for i5/OS SQL Performance Workshop
    • ibm.com/systemi/db2/db2performance.html
    • ibm.com/partnerworld/wps/training/i5OS/courses
  – Indexing & Stats Strategy White Paper
    ibm.com/servers/enable/site/education/ibo/record.html?indxng
  – Database modernization roadmaps
    • Modernizing DB2 definitions and usage
      http://www.developer.ibm.com/vic/hardware/myportal/develop/roadmap?roadMapId=appiniti
    • Modernizing data access with SQL
      http://www.developer.ibm.com/vic/hardware/myportal/develop/roadmap?roadMapId=appinitj
    • Optimizing SQL performance
  • Identify First Project
    – Write a new function/program component using SQL
    – Rewrite an existing component using SQL (ie, reporting function)
    – Port SQL-based program to DB2 for i5/OS
      • Porting guides & conversion tools at: ibm.com/servers/enable/site/db2/porting.html
  • Get Help
    – Lab Services Technology Center - ibm.com/systems/services/labservices/

The New Power Equation
**Additional Information**

- **DB2 for i5/OS Websites**
  - Homepage: ibm.com/systemi/db2
  - developerWorks Zone: ibm.com/developerworks/db2/products/db2i5OS

- **Newsgroups**
  - USENET: comp.sys.ibm.as400.misc, comp.databases.ibm-db2

- **Education Resources - Classroom & Online**
  - http://ibm.com/systemi/db2/db2educ_m.html
  - http://ibm.com/partnerworld/wps/training/i5OS/courses

- **DB2 for i5/OS Publications**
  - Porting Help: http://ibm.com/servers/enable/site/db2/porting.html
  - DB2 for i5/OS Redbooks (http://ibm.com/systemi/db2/reldrebooks.html)
    - Stored Procedures, Triggers, & User-Defined Functions on DB2 for iSeries (SG24-6503)
    - DB2 for AS/400 Object Relational Support (SG24-5409)
    - Advanced Functions & Administration on DB2 for iSeries (SG24-4249)
    - OnDemand SQL Performance Analysis … in V5R4 (SG24-7326)
    - Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS (SG24-6598)
  - SQL/400 Developer's Guide by Paul Conte & Mike Cravitz
IBM DB2 for IBM i Consulting and Services

✓ Database modernization (Look for an upcoming DB2 for i modernization workshop)
✓ DB2 Web Query
✓ Database design, features and functions
✓ DB2 SQL performance analysis and tuning
✓ Data warehousing review and assessment
✓ DB2 for i education and training

Contact: Tom McKinley
mac2@us.ibm.com
IBM Systems and Technology Group
Rochester, MN USA
Trademarks and Disclaimers

© IBM Corporation 1994-2009. All rights reserved.
References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Cell Broadband Engine and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and are used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.
Information is provided "AS IS" without warranty of any kind.
The customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.
All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.
Prices are suggested U.S. list prices and are subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

The New Power Equation