

ibm.com



iSeries. mySeries.

Application Modernization DB2 UDB Style

Mike Cain

BP11

IBM eServer iSeries
ITSO Technical Forum
2004



Copyright IBM Corporation 2004

Agenda

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

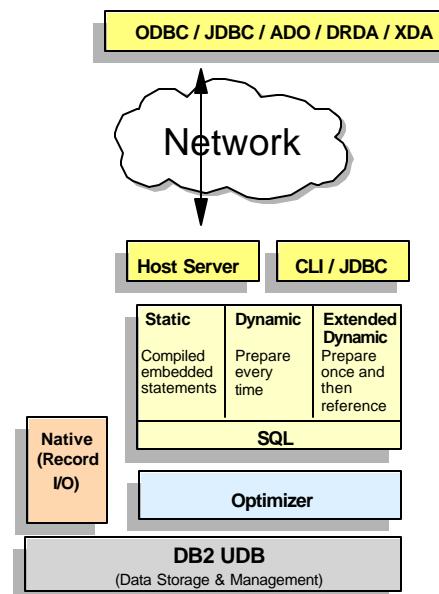
Copyright 2004 - IBM Corporation

Why Structured Query Language (SQL)?

- Portability of code & skills
 - Strategic database interface for industry & OS/400
 - ▶ 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 iSeries as a Database Server
 - SQL as a programming language can reduce Total LOC
 - DB2 SMP - parallel database processing
-

Copyright 2004 - IBM Corporation

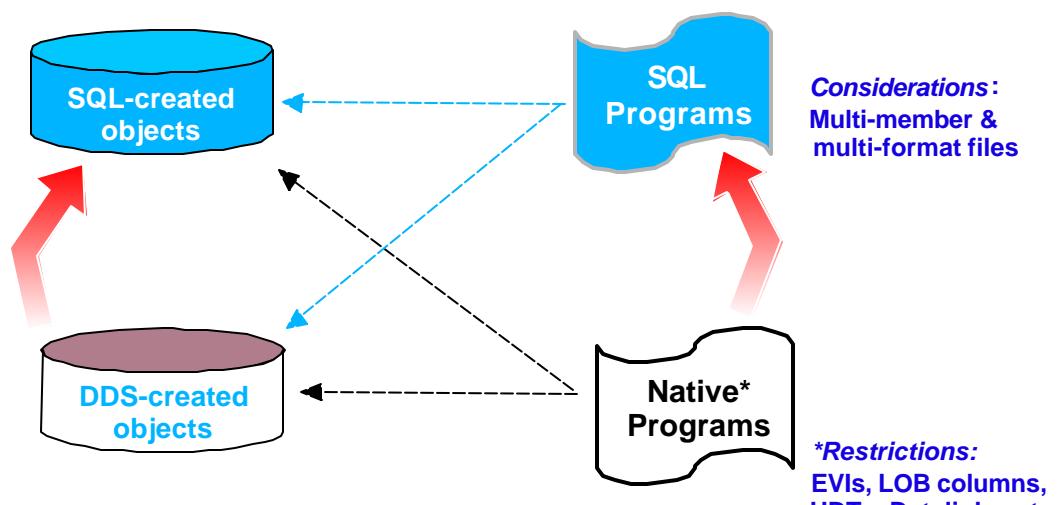
Approaches & Options



Copyright 2004 - IBM Corporation

- High level picture of DB architecture and where the optimization occurs
- ADO = Active data objects (i.e. OLE DB)
 - Implemented via ODBC or directly to Host Server (project Lightning)
- ODBC/JDBC/ADO = client query program interfaces
- CLI/JDBC = server query program interfaces
- All the components will be covered throughout the course
- Components will be covered from the bottom up

Approaches & Options



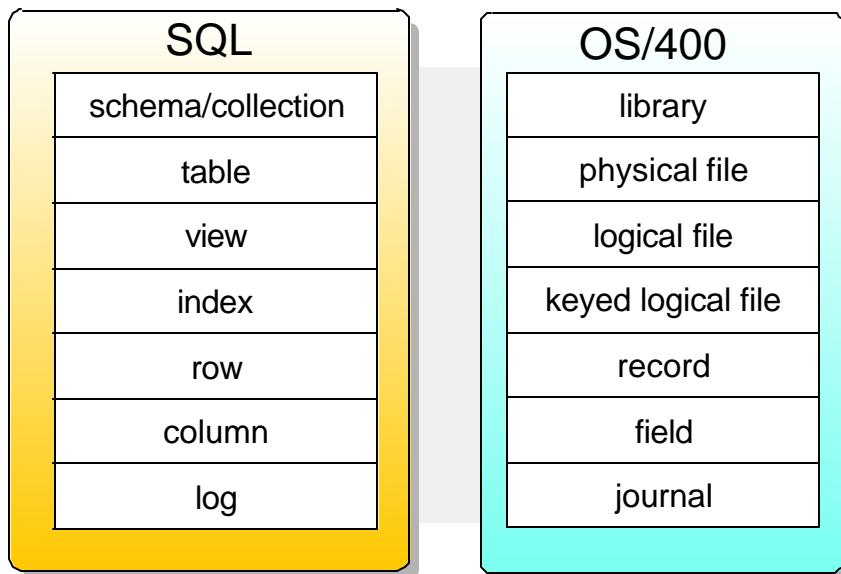
Copyright 2004 - IBM Corporation

Modernizing Definitions & Objects

- Terminology
- Moving from DDS to SQL DDL
- SQL object management
- Embedding business logic into database definitions

Copyright 2004 - IBM Corporation

Modernizing Database Objects - Terminology



Copyright 2004 - IBM Corporation

- ▶ The terms on the left side of the table equate to the terms on the right side. Traditional OS/400 users are accustomed to using the terms on the right. While the terms used may be different, the function provided is the same.
- ▶ You'll notice that member is not anywhere on the chart, no SQL equivalent

Modernizing Database Definitions & Objects

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)

Copyright 2004 - IBM Corporation

- ▶ One disadvantage used to be that SQL tables did not support field reference files - fixed with V5R2 CREATE TABLE LIKE enhancement

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)
 - Change was made to improve the performance of queries that scan lots of key values in an index (64K I/O operation brings more keys into memory)
 - Single key lookups in index may or may not be as efficient
 - Larger logical page size can lend itself to more efficient index maintenance
 - Considerations:
 - ◆ Larger index pages may strain memory-starved environments
 - ◆ SQL Indexes that are journaled explicitly or implicitly (SMAPP) will increase size of journal receivers
 - > Use RCVSIZOPT(*RMVINTENT *MAXOPT2)
 - > Journal Receiver Threshold should be at least 6.5 GB
 - ▶ Disadvantages
 - 8K Logical Page Size
 - No support for Select/Omit filtering or join logical files

Copyright 2004 - IBM Corporation

- ▶ SQL Indexes can be opened as keyed logical files - SQL Index "DDS" includes all of the fields into the logical file

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 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

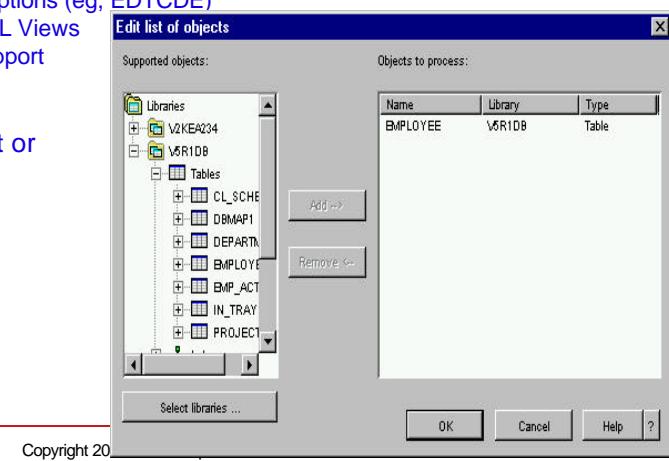
Copyright 2004 - IBM Corporation

Modernizing Database Definitions & Objects

DDS to SQL Conversion Tool

■ **iSeries Navigator Generate SQL Task (QSQGNDDL 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 unconvertable options (eg, 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



Copyright 20...

Modernizing Database Definitions & Objects

SQL Object management

- **SQL Source Management best practices:**

- ▶ Just like DDS SQL source can be stored in source physical file members just and referenced with the RUNSQLSTM CL command instead of CRTPF/CRTLF
 - If change management tools are not iSeries specific, store SQL scripts in PC or IFS files
 - If SQL source misplaced, Generate SQL can be used to retrieve the SQL source from System Catalogs (SYSIBM & QSYS2)

- ▶ SQL Table definitions can use Field Reference File

```
CREATE TABLE customer AS  
  (SELECT id cust_id, lname 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

Copyright 2004 - IBM Corporation

- ▶ One disadvantage used to be that SQL tables did not support field reference files - fixed with V5R2 CREATE TABLE LIKE enhancement
 - ▶ SQL Field Reference support forces you to use all data types from field reference file

Modernizing Database Definitions & Objects

SQL Object Management

- **SQL Column & Object names have maximum lengths of 30 & 128, but many OS/400 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 >> CUSTO00001
- **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 iSeries SQL syntax to specify your own short name**
 - ▶ RENAME TABLE (tables & views) & RENAME INDEX
 - ▶ FOR COLUMN clause for columns
 - ▶ SPECIFIC clause for procedures, functions

Copyright 2004 - IBM Corporation

- ▶ One disadvantage used to be that SQL tables did not support field reference files - fixed with V5R2 CREATE TABLE LIKE enhancement

Modernizing Database Definitions & Objects

SQL Object Management

■ 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
```

Copyright 2004 - IBM Corporation

- ▶ One disadvantage used to be that SQL tables did not support field reference files - fixed with V5R2 CREATE TABLE LIKE enhancement

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

Copyright 2004 - IBM Corporation

Modernizing Definitions & Objects

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 )
```

Copyright 2004 - IBM Corporation

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)
```

Copyright 2004 - IBM Corporation

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 is independent of the user interface
 - ▶ Can be used to transform data before it gets into DB2
- DB2 UDB for iSeries 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

```
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)
```

Copyright 2004 - IBM Corporation

Modernizing Data Access

- **Programming Interfaces**
- **Native I/O to SQL Comparison**
- **Using SQL to Reuse & Repurpose Existing Code**
- **DB2 & New Application Models**
- **Tools**

Copyright 2004 - IBM Corporation

Modernizing Data Access

Programming Interfaces

Static SQL	Dynamic SQL	Extended Dynamic SQL
Embedded Static	Embedded Dynamic	QSQPRCED
	JDBC, SQLJ	Toolbox JDBC & iSeries Access ODBC
	OLE DB, .NET	XDA API set
	CLI, ODBC	
	Net.Data	
	RUNSQLSTM	

*****DB2 UDB SQL Development Kit only required if embedded SQL into HLL programs***

Copyright 2004 - IBM Corporation

Modernizing Data Access

Native I/O to SQL Example

```

...
C/EXEC SQL
C+ DECLAREsqlJn 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   SearchKey  KList
C   Kfld       Kfld
C   ::          Times   Occur   Result_Set
C   SearchKey  Setll   If      TIME_DIML1
C   If          DOU    READ   %FOUND
C   If          LEAVE  EOF    RowsReq = RowsRd
C   Endif      DATEKEY Setll   ITEMFACTL1
C   If          If      %FOUND
C   If          DOU    READE  ITEMFACTL1
C   If          LEAVE  EOF    %EOF
C   Endif      PARTKEY CHAIN  PART_DIML1
C   If          If      Not %FOUND
C   Iter
C   Endif      CUSTKEY CHAIN  CUST_DIML1
C   If          If      Not %FOUND
C   Iter
C   Endif      SUPPKEY CHAIN  SUPP_DIML1
C   If          If      Not %FOUND
C   Iter
C   Endif ...

```

Copyright 2004 - IBM Corporation

- SQL only getting a subset of the columns, RPG/Native gets all of the fields back - if you want to change "projection", then need to create a new logical

Modernizing Data Access

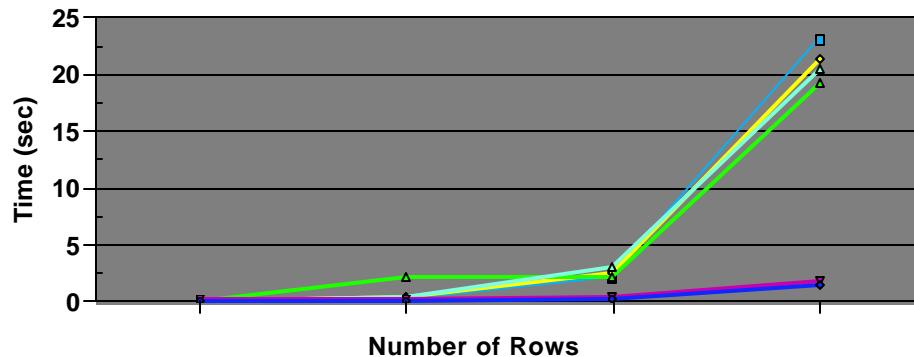
Native I/O to SQL Example - Joined LFs & Views

```
...  
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   Occur      Result_Set  
C   READ      NTVJOIN002  
C   If          %EOF  
C   Leave      Endif  
C   Endif  
C           Eval  RowsRd = RowsRd + 1  
C           ENDDO  
C           Endif
```

Copyright 2004 - IBM Corporation

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

	1	100	1000	10000
Native File Join	0.002512	0.260248	2.219504	23.228176
Native JoinLF	0.002304	0.362128	2.544608	21.366480
Native JoinLF w/blk	0.002400	0.144288	2.125032	19.311464
SQL - No IOA	0.145160	0.489136	3.166704	20.452984
SQL IOA	0.251168	0.267208	0.417800	1.898800
SQL SQE IOA	0.013536	0.019320	0.250160	1.576536

Copyright 2004 - IBM Corporation

- ▶ All SQL runs did blocked fetches
- ▶ Findings
 - ▶ RLA slightly better than SQL when retrieving small number of rows
 - ▶ 1 to 100 rows
 - ▶ SQL outperforms RLA when retrieving large result sets
 - ▶ ~10000 rows or more
 - ▶ Blocking DDS adds more up front overhead
 - ▶ more efficient for larger volumes
 - ▶ similar to SQL support which implicitly blocks
 - ▶ Index only access provides dramatic performance efficiency at approximately 100 rows
 - ▶ available to SQL only
 - ▶ SQE optimization costs are getting closer to RLA costs
- ▶ Conclusions
 - ▶ In general, when processing sets of data, SQL performance is equivalent to or better than RLA access
 - ▶ RLA outperforms SQL for single or small numbers of row access. However, tuning SQL via Index only access may achieve run time or response time objectives for small result sets.
 - ▶ For mission critical applications RLA may be the best choice
 - ▶ Continue to revisit SQL with each release

Modernizing Data Access

Comparing SQL & Native Performance

	RPG	SQL
<i>Arrival Sequence</i>	202,000	173,000
<i>Keyed Access</i>	131,000	117,000
<i>Keyed Access (memory-constrained)</i>	8000	12,000
<i>Pseudo-Random Keyed Access</i>	3950	3440
<i>Direct Key Access</i>	5100	4100

rows per minute

Published by *NEWS/400* in March 2000 - Measured on V4R4
V5R2 and V5R3 DB Engine is faster

Copyright 2004 - IBM Corporation

- ▶ Highly optimized RPG code with specialized override
- ▶ Southern Wine:
 - ▶ 24 hrs vs minutes

Modernizing Data Access

Native to SQL Considerations

- ORDER BY clause is 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 - still missing from RPG Precompiler in latest release:
 - ▶ Support for free format embedded SQL
 - ▶ Support for properly scoping a local variable in a subprocedure as a host variable in an SQL statement (ie, multi-pass)
 - ▶ 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 ODPs (Open Data Paths)
 - Prepare Once, Execute Many
 - Connection Pooling
 - Keep Connections & Jobs active as long as possible
 - ▶ Blocked Fetches & Inserts

Copyright 2004 - IBM Corporation

- ▶ Websphere EJBs require an isolation level other than *NONE

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 UDB for iSeries support provides maximum flexibility by supporting both SQL & External types**
 - ▶ External support allows reuse of existing iSeries code & skills
 - ▶ SQL Procedural Language (PSM) enables better portability of logic (& programming skills) to/from other platforms
 - ▶ Data security can be enhanced/maintained with OS/400 Adopted Authority

Copyright 2004 - IBM Corporation

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:

```
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...
```

Copyright 2004 - IBM Corporation

Modernizing Data Access

Using SQL to Reuse & Repurpose Existing Code: Procedures

- **Stored Procedures** are similar to UDFs, but invoked with an SQL Call statement
 - ▶ More than just a remote program call
 - Supported by majority of the application development tools & languages
 - Result sets can improve performance by minimizing network trips
 - ▶ Procedures most effective when multiple operations performed on a single procedure call
 - ▶ External Procedure considerations
 - May have to make slight modifications to existing code to match stored procedure parameter conventions or develop "wrapper" procedures
 - Need to design process for installing & upgrading stored procedures
 - ▶ SQL Procedure considerations
 - Generated C code with embedded SQL not as fast as user-written code
 - Minimize the number of nested calls to other SQL procedures
 - No support for blocked fetches & inserts

Copyright 2004 - IBM Corporation

Modernizing Data Access

DB2 & New Application Models - Distributed Transactions

- **DB2 UDB for iSeries supports XA Transactions**

- ▶ OS/400 Transaction manager was substantially enhanced in V5R2 for distributed transaction modes - including support for JTA
- ▶ OS/400 XA support designed for SQL Interfaces
- ▶ Native code can participate in XA Transactions if they are "wrapped" as external stored procedures or user-defined functions

Copyright 2004 - IBM Corporation

Modernizing Data Access

DB2 & New Application Models - Web Services

■ DB2 Web Services - WORF (Web Services Object Runtime Framework)

- ▶ Easily create simple XML based Web services that access DB2 - part of Websphere Express
- ▶ Create DADX documents by using a simple text editor, and deploy them in Websphere with minimal knowledge of XML or SQL
- ▶ Used through Websphere and SOAP
 - SQL Select, Update, Insert, Delete, Procedures, etc.
 - XML Extender can also be used

▶ Read more at:

www.iseriesnetwork.com/artarchive/index.cfm?fuseaction=viewarticle&CO_ContentID=17566&channel=art&subart=auth&authid=329

```
<?xml version="1.0"?>
<ADX xmlns="http://schemas.ibm.com/db2/dxx/dadx"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl">Accesses the city versus zip code database.
  </documentation>
  <operation name="findCityByZipCode">
    <documentation xmlns="http://schemas.xmlsoap.org/wsdl"/>Finds the city for a zip code.
    </documentation>
    <query>
      <SQL_query>SELECT CITY FROM DEMO.CITY_TAB WHERE ZIPCODE = :zipcode</SQL_query>
      <parameter name="zipcode" type="xsd:string"/>
    </query>
  </operation>
</ADX>
```

Copyright 2004 - IBM Corporation

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
- **Text Mining with DB2 Text Extender**
 - ▶ 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

Copyright 2004 - IBM Corporation

Modernizing Data Access

Development Tools

- **WebSphere Development Studio Client for iSeries**
 - ▶ DB2 Web Service Support
 - ▶ XML Extender Aids
 - ▶ SQL statement wizard
- **RSE & Code/400 co-exist with SQL**
- **PDM has basic SQL prompting support**
- **Visual Studio .NET Plugins** (Beta - <https://www6.software.ibm.com/reg/dm/dm-adtpapp-i>)
- **DB2 Information Integrator for non-DB2 data access**
ibm.com/servers/enable/site/education/ibo/record.html?hetdata
- **DB2 Development Center**
 - ▶ Stored Procedures
 - ▶ User-Defined Functions

Copyright 2004 - IBM Corporation

Modernizing Data Access

Development Tools

- **Toolbox Graphical Debugger for ILE & SQL-source level debug**
[\(ibm.com/servers/enable/site/education/abstracts/sqldebug_abs.html\)](http://ibm.com/servers/enable/site/education/abstracts/sqldebug_abs.html)
- **iSeries Navigator**
 - ▶ Editors for procedure, functions, triggers
 - ▶ SQL statement wizard for INSERT, SELECT, UPDATE
 - ▶ Downloadable Tutorials at:
[\(ibm.com/servers/enable/education/i/ad/db2/recentindex1.html\)](http://ibm.com/servers/enable/education/i/ad/db2/recentindex1.html)
- **Visual Explain**
- **Database Monitor (STRDBMON CL command)**
- **PRTSQLINF**
- **DB2 SMP - licensed feature (OS/400 Option 26)**

Copyright 2004 - IBM Corporation

Next Steps

■ EDUCATION

- ▶ iSeries SQL Performance Workshop
 - ibm.com/servers/eserver/iseries/service/igs/db2performance.html
 - ibm.com/servers/enable/education/i/ad/db2/recentindex1.htm
- ▶ Indexing & Stats Strategy White Paper
ibm.com/servers/enable/site/education/ibo/record.html?indxng

■ 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 UDB for iSeries
 - Porting guides & conversion tools at: ibm.com/servers/enable/site/db2/porting.html

■ Get Help

- ▶ Solutions Enablement DB2 Technology Team
- ▶ eServer Custom Technology Center - www.ibm.com/ctc
- ▶ iSeries IBM Global Services Team - heithoff@us.ibm.com

Copyright 2004 - IBM Corporation



DB2 UDB Family Certifications



Certified Database Associate - DB2 UDB Family (Test 700)

- **Website:** ibm.com/certify/certs/dbdaudv81.shtml
- **Education Resources:** ibm.com/certify/tests/edu700.shtml
- **Online Tutorial:**
www7b.boulder.ibm.com/dmdd/library/tutorials/db2cert/db2cert_V8_tut.html

Certified Application Developer - DB2 UDB Family (Test 703)

- **Website:** ibm.com/certify/certs/dbapudv81.shtml
- **Education Resources:** ibm.com/certify/tests/edu703.shtml

➤ **Sample Tests:** certify.torolab.ibm.com/ice

➤ Exams were refreshed & updated for DB2 UDB for iSeries

Additional Information

■ **DB2 UDB for iSeries home page** - <http://www.iseries.ibm.com/db2>

■ **Newsgroups**

- ▶ USENET: comp.sys.ibm.as400.misc, comp.databases.ibm-db2
- ▶ iSeries Network (NEWS/400 Magazine) SQL & DB2 Forum -
<http://www.iseriesnetwork.com/Forums/main.cfm?CFApp=59>

■ **Education Resources - Classroom & Online**

- ▶ http://www.iseries.ibm.com/db2/db2educ_m.htm
- ▶ <http://ibm.com/servers/enable/education/i/ad/db2/recentindex1.html>

■ **DB2 UDB for iSeries Publications**

- ▶ Online Manuals: <http://www.iseries.ibm.com/db2/books.htm>
- ▶ Porting Help: <http://ibm.com/servers/enable/site/db2/porting.html>
- ▶ DB2 UDB for iSeries Redbooks (<http://ibm.com/redbooks>)
 - [Stored Procedures & Triggers on DB2 UDB for iSeries](#) (SG24-6503)
 - [DB2 UDB for AS/400 Object Relational Support](#) (SG24-5409)
 - [Advanced Functions & Administration on DB2 UDB for iSeries](#) (SG24-4249)
- ▶ **SQL/400 Developer's Guide** by Paul Conte & Mike Cravitz
 - <http://as400network.com/str/books/Uniquebook2.cfm?NextBook=183>

Copyright 2004 - IBM Corporation

Trademarks & Disclaimers

© IBM Corporation 1994-2004. 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.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM(logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.

Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All 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 in this presentation 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. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation 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.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Copyright 2004 - IBM Corporation