

Table of contents

Course overview	2
Agenda	Error! Bookmark not defined.
DB2 UDB for iSeries strategic initiatives	2
Application development enhancements	4
Integration of free-format RPG and SQL	5
iSeries Access for Windows	6
DB2 Connect Unlimited Edition for iSeries	7
Application development	8
Application development (continued)	9
Stored procedures	10
SQL and DB2 UDB enhancements	11
OLAP expressions (ROW_NUMBER)	12
OLAP expressions (RANK and DENSE_RANK)	13
Recursive SQL	14
Recursive SQL example (initialization phase)	15
Recursive SQL example (recursion pass #1)	16
Recursive SQL example (recursion pass #2)	17
Recursive SQL example (final results)	18
Recursive SQL considerations	19
Recursive SQL considerations (continued)	20
SELECT advancements	21
Richer toolbox of SQL functions	22
Performance improvements	25
Performance enhancements	26
Significant XML extender enhancements	27
SQL enhancements available only to SQE	28
Enhanced performance tooling	29
DB2 UDB family and MQTs (stage 2 i5/OS V5R4)	31
System i5 platform optimized for SAP	32
Availability and recovery enhancements	33
More robust catalogs and cross-reference files	34
Journaling enhancements	35
Ease-of-use and management improvements	36
iSeries Navigator enhancements	37
Enhanced index advice	38
Live DB2 UDB performance analysis via SQL plan cache	39
Advanced database monitor filtering	41
Simplifying database monitor analysis	42
Simplifying database monitor analysis (continued)	43
Additional management enhancements	44
DB2 Content Manager Standard Edition V8.3	45
Summary	47
Resources	48
Trademarks and special notes	51

Instructions

Requirements

- JavaScript is required to navigate
Click [here](#) to test your browser for compatibility.

Browsers

If you have one of these browsers you will be set.

- Netscape Communicator/Navigator Version 4.0.5 or better
- Microsoft Internet Explorer Version 4.0 or better
- Mozilla Firefox

You can download this course and take it off-line

This course can be downloaded so you can conveniently take it off-line. You will need to have installed a copy of Adobe Acrobat Reader (you can download this reader by clicking [here](#)). If you want to download the Acrobat files related to this course click here. There are two separate files to download. One contains the foil screens (graphics), the other contains the speaker notes (text). The foil and speaker notes pages have been numbered to help you in putting them together.

Resources

As you go through this course, you will observe, and probably take advantage of, frequent hotlinks that send you to other Web sites for more details about a particular topic. These hotlinks have been compiled in one long list at the end of the course (under **Resources**) and can be accessed by clicking on the LINKS option in the navigator bar.

We think these links are important, but you might prefer to visit these sites at some later point so that, for right now, you can stay focused on the “meat” within this course. You can clip the contents of this Resources page into a document file on your local PC drive. Then, at your convenience, you can use these saved hotlinks to explore these topics in greater detail.

The Resources page also has references pertaining to this course.

Speaker Notes

Open or close the Speaker Note window with these buttons:

Navigation

These are the general navigation buttons:

- Click NEXT to go forward one page.
- Click PREV to go back one page.
- The AGENDA page lists all the pages in this IBE for quick navigation.
- The LINKS page contains the hotlinks to related information.
- The GLOSSARY page contains definitions of terminology used in this course.
- The FAQ page lists frequently asked questions about the subject.
- Click EXIT to leave the course.



Kent Milligan

DB2 UDB technology specialist
IBM ISV Business Strategy and Enablement

Kent Milligan is a DB2 UDB technology specialist in IBM ISV Business Strategy and Enablement. Kent spent many years of his IBM career as a member of the DB2 UDB development group in Rochester, Minnesota. He speaks and writes regularly on various System i5 relational database topics.



System and Technology Group

DB2 UDB for iSeries: V5R4 Overview

Kent Milligan – kmill@us.ibm.com

© 2006 IBM Corporation

Course overview

DB2 UDB for iSeries strategic initiatives

Openness (industry standard)

Support

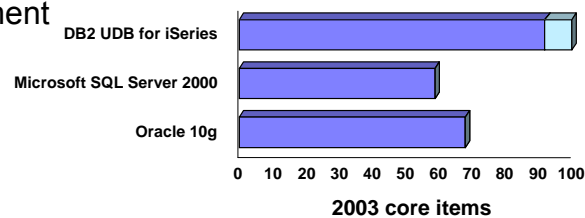
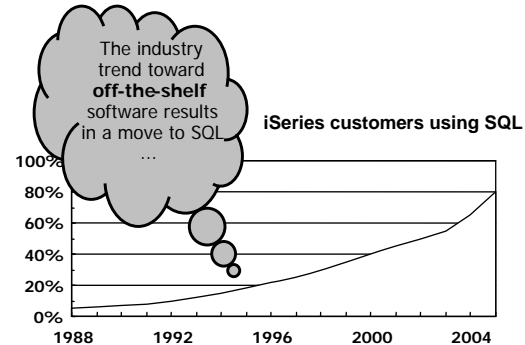
- Accommodate solution providers
- Portability and compatibility
- Flexibility

Continued LEADERSHIP in database technologies

- Consistency across DB2 UDB family
- Shared research and development across IBM labs

Continued leveraging of System i5 strengths

- Availability
- Scalability
- Usability (total cost of ownership)
- Application flexibility



**SQL 2003 core standard
100% complete with
DB2 UDB for iSeries V5R4**

DB2 UDB for iSeries strategic initiatives

This chart shows the strategic direction IBM has established for DB2 UDB for iSeries. As you progress through this course, you will be able to relate the DB2 UDB for iSeries V5R4 enhancements back to these initiatives.

SQL is the industry standard for database access and programming. Although the heritage of application development on the System i5 platform has been to use native interfaces, more and more application developers are using SQL as a programming language. This is the result of critical IBM investments that support SQL technologies in the DB2 UDB for iSeries product. In the upper graph on this chart, you can see the trend toward greater use of SQL. SQL is the strategic database interface for DB2 UDB for iSeries to accommodate the porting of third-party applications and tools, almost all of which utilize SQL-based interfaces for their data access.

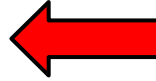
As you can also see on the lower graph, DB2 UDB for iSeries actually is one of the first major databases to comply fully with the core level of the SQL 2003 Standard. The beauty of the DB2 UDB family is that every implementation, on all IBM System platforms, benefits from this research. DB2 UDB for iSeries will continue to leverage this technology leadership position and will maintain close compatibility with the other DB2 UDB family members.

DB2 UDB for iSeries: V5R4 Overview

Additionally, DB2 UDB for iSeries will sustain its record of leveraging and building on the strengths of the System i5 platform. These strengths include: reliability, availability, total cost of ownership, scalability, and usability.

Agenda

- **Application development enhancements**
- **SQL and DB2 UDB enhancements**
- **Performance improvements**
- **Availability and recovery enhancements**
- **Ease-of-use and management improvements**



Application development enhancements

DB2 UDB for iSeries V5R4 is packed with new and enhanced application development functions: integration of free-format RPG and SQL, enhanced support for various programming interfaces, including the SQL call level interface (CLI) standards, and many other functions. This section discusses these in detail.

Integration of free-format RPG and SQL

```

d getOrderCount    pi          10i 0
d  inCustNumber    9b 0 const
d  outSqlState     5a
d  outSqlMsg       256a

d  ordercount      s          10i 0

/free
  outSqlMsg = *blanks;
  exec sql SELECT count(*) into :orderCount FROM orders
        WHERE cust_no = :inCustNumber;

  outSqlState = sqlState;
  if %subst(sqlState:1:2) <> '00';
    exec sql GET DIAGNOSTICS CONDITION 1
          :outSqlMsg = MESSAGE_TEXT;
    orderCount = 0;
  endif;
  return orderCount;
/end-free

```

Integration of free-format RPG and SQL

Many RPG developers have improved their productivity in recent years with the usage of the free-form syntax. DB2 UDB for iSeries V5R4 brings even greater productivity by allowing you to embed SQL requests into free-format RPG as demonstrated in the red code shown in this example. This ability to integrate SQL in free-format style is a first for RPG programmers.

iSeries Access for Windows

- **.NET provider enhancements**

- LOB column support**
- System naming and library list support**
- MS FW 2.0 compatibility**
- Boost Intellisense support to aid programmers
- Multiple active result sets on a connection
- Customizable string processing

**Available with latest i5/OS V5R3 service pack

- **JDBC (Version 3 currency and performance)**

- Optimization goal connection attribute

- **IBM Enterprise Workload Management (EWLM) support added to:**

- CLI, DRDA, .NET, ODBC, and JDBC

- **Driver support for Windows Vista**

- Depending on availability

- **OLE database driver**

- System naming and library list support

- **ODBC driver**

- Optimization goal connection attribute

iSeries Access for Windows

In addition to embedded SQL, many iSeries developers use industry-standard interfaces, such as Java Database Connectivity (JDBC) and Microsoft® .NET technologies, to build innovative solutions. Enhancements to these programming interfaces that are part of iSeries Access for Windows include customization for better integration of specific iSeries capabilities along with keeping current with industry standards.

IBM Enterprise Workload Manager (EWLM) allows you to define business-oriented performance goals for an entire domain of servers. EWLM then provides an end-to-end view of actual performance measurements that are relative to those goals. With EWLM, you can ensure that work requests (executed across a mixed-platform, multi-tiered application environment) perform according to your expectations. The DB2 UDB for iSeries V5R4 enhancements make it easy to instrument your DB2 UDB for iSeries workloads for EWLM.

DB2 Connect Unlimited Edition for iSeries

- **New customized package for iSeries developer community (available since 3rd quarter 2005)**
- **Affordable access to:**
 - DB2 UDB add-ins for Visual Studio
 - Federated Data access
 - Joining of DB2 UDB tables in different iSeries partitions
 - Joining of DB2 UDB tables across iSeries and Linux, UNIX, and Windows servers
 - Extend iSeries data to mobile devices: DB2 Mobility on Demand
- **More details:**
 - ibm.com/software/data/db2/db2connect/edition-uei.html

DB2 Connect Unlimited Edition for iSeries

The recent introduction of IBM DB2 Connect™ Unlimited Edition for iSeries also makes DB2 UDB plug-ins for Microsoft Visual Studio® more accessible to .NET developers who want to process DB2 UDB for iSeries databases. DB2 Connect Unlimited Edition is a special version of the IBM DB2 Connect offering that IBM has packaged and priced for the iSeries marketplace.

In addition to the .NET plug-ins, this product also provides heterogeneous access to database servers (other than DB2 UDB) with federated database support and the ability to make DB2 UDB for iSeries data available on mobile devices.

This DB2 Connect offering simplifies application development with Web services support and provides a portable interface for applications to access any of the DB2 UDB family members. DB2 Connect Unlimited Edition does not run natively on the i5/OS platform; it connects to DB2 UDB for iSeries from Microsoft Windows®, UNIX®, and Linux™ clients. You can access more details about this new product at the DB2 Connect Unlimited Edition Web site listed in the **Resources** section of this course.

Application development

▪ CLI enhancements

- Maximum handles limit doubled to 160,000
- **SQLFetchScroll**, **block-fetch** and **column-wise** binding
- Column-wise blocked insert binding
- Optimization goal connection attribute
- Cursor sensitivity statement attribute
- Improved XA documentation
- New **SQLGetInfo** and **SQLColAttributes** options
- Max rows attribute (**SQL_ATTR_MAX_ROWS**)

▪ XA over DRDA

▪ Redesigned SQL descriptor area (SQLDA)

- Support for longer column names
- Faster internal processing

Application development

The SQL call level interface (CLI) is a popular server-side programming interface that many applications use for data access. An increased handle limit (from 80,000 to 160,000) and new connection and statement attributes give developers greater flexibility when enhancing their applications.

Extending XA support to the Distributed Relational Database Architecture (DRDA) allows developers to take advantage of distributed transactions through DRDA-based drivers, such as the universal JDBC driver within DB2 Connect Unlimited Edition for iSeries.

All dynamic SQL interfaces (such as CLI, JDBC, and .NET) directly or indirectly use the SQL descriptor area (SQLDA) to manage input and output data. With DB2 UDB for iSeries V5R4, IBM redesigned the internal structures and processing for SQLDA to be more efficient. This internal redesign has resulted in a nice performance boost for dynamic SQL applications in DB2 UDB for iSeries V5R4 (without any programmer intervention).

Application development (continued)

- **Bigger and better limits**
 - 2-MB SQL statement maximum
 - 1,000 tables per query
 - 128-byte column names
 - 1024 parameters for stored procedure
 - 32-KB index keys and ORDER BY statement
- **New SET SESSION authorization statement**
 - Better SQL auditing with the ability to supply actual user of remote connections
 - New SESSION USER and SYSTEM USER special registers
 - Considerations:
 - Requires *ALLOBJ authority to execute
 - Frees DB2 UDB for iSeries resources, which can impact performance
 - Might require resetting other settings (such as SQL Path)

Application development (continued)

SQL programmers can now be more descriptive with their column names because the maximum size of the column name is larger (128 bytes, instead of its former 30 bytes). Programmers can embed more complicated business logic into their SQL statements by referencing up to 1,000 tables and by coding statements with lengths up to 2 megabytes. The previous maximum statement length was 64 kilobytes. (This increase will be most beneficial to programmers who create SQL procedures, functions, and triggers).

Many distributed and Web-based applications utilize connection pooling to improve performance. Often, applications set up pooled connections to use a shared, generic user ID for establishing the connection. The result is that, when monitoring data access, most diagnostic tools will only report the shared user ID instead of the real user's identity. With i5/OS V5R4, you can code the new **SET SESSION** authorization statement to execute after accessing a pooled connection to identify the actual identity of the end user. Though this makes database auditing more accurate, the downside is that it is not possible to reuse many DB2 UDB resources, such as open data paths (ODPs), when the user ID for a connection changes. Thus, executing this new statement can negate some of the performance benefits of using a pooled connection. DB2 UDB V5R4 includes new special registers that allow an application to retrieve the current session user ID, as well as the shared user ID that was used to set up the connection with the system user register.

Stored procedures

- DB2 UDB expression evaluator for **faster** SQL procedural language (existing objects must be recreated)
- Easier authority setup with support for DFTRDBCOL and DYNDFTCOL attributes
 - Useful when porting from other database management systems
 - Also available on OS/400 V5R2 and i5/OS V5R3 with latest database group PTF
- Simpler maintenance with new **ALTER PROCEDURE** statement


```
ALTER PROCEDURE Increase_Level
REPLACE
BEGIN
  DECLARE CurLvl INT;

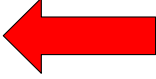
  SELECT edlevel INTO CurLvl FROM emptbl
    WHERE empno=Emp#;
  IF NwLvl > CurLvl THEN
    UPDATE emptbl SET edlevel=NwLvl, salary=salary + (salary*0.10)
      WHERE empno=Emp#;
  END IF
END
```
- Plans to support debug environment for DB2 UDB Development Center

Stored procedures

Improved performance is the main DB2 UDB for iSeries V5R4 highlight for SQL stored procedures. When the DB2 UDB engine creates an SQL procedural object, it generates C code with embedded SQL to implement the business logic defined with the procedural SQL statements. You will probably agree that generated code almost never performs as well as code that you write yourself. This has been true for SQL procedures since their introduction in IBM OS/400® V4R2. i5/OS V5R4 makes a significant change to the DB2 UDB C code generation engine for the SQL procedural language. Obviously, the improvements on your system will vary, but some tests in the lab showed a 30% performance improvement. You will need to recreate all SQL procedures, triggers, and functions on DB2 UDB for iSeries V5R4 to benefit from this enhancement.

The **ALTER PROCEDURE** statement simplifies the ongoing development and maintenance of SQL procedures. This new statement enables you to change the attributes or logic of an existing procedure quickly without the need either to delete the existing procedure first or to reset the ownership and privileges for the procedure object. In the example at the bottom of this chart, the **ALTER PROCEDURE** statement changes the **Level_Increase** logic to grant a 10% salary increase instead of the 5% increase that the original procedure implemented.

Agenda

- Application development enhancements
- SQL and DB2 UDB enhancements 
- Performance enhancements
- Availability and recovery enhancements
- Ease-of-use and management improvements

SQL and DB2 UDB enhancements

In this next section, you will learn about new OLAP expressions, sophisticated recursive SQL processing, improved **SELECT** statements, more robust SQL functions in the IBM Toolbox for Java™, a new trigger type, and other enhancements.

OLAP expressions (ROW_NUMBER)

- **ROW_NUMBER** — use to arbitrarily assign a number to rows in the final result set

**SELECT ROW_NUMBER() OVER (ORDER BY workdept, lastname) AS Nbr,
lastname, salary FROM employee ORDER BY workdept, lastname**

NBR	LASTNAME	SALARY
1	HAAS	52750.00
2	HEMMINGER	46500.00
3	LUCCHESSI	46500.00
4	O'CONNELL	29250.00
5	ORLANDO	29250.00

**SELECT workdept, lastname, hiredate,
ROW_NUMBER() OVER (PARTITION BY workdept ORDER BY hiredate) AS Nbr
FROM employee
ORDER BY workdept, hiredate**

WORKDEPT	LASTNAME	HIREDATE	NBR
A00	LUCCHESSI	1958-05-16	1
A00	O'CONNELL	1963-12-05	2
A00	HAAS	1965-01-01	3
A00	HEMMINGER	1965-01-01	4
A00	ORLANDO	1972-05-05	5
B01	THOMPSON	1973-10-10	1
C01	QUINTANA	1971-07-28	1
C01	KWAN	1975-04-05	2

OLAP expressions (ROW_NUMBER)

iSeries programmers have often wanted the ability to arbitrarily number the rows that a query returns in order to have an easy way of identifying the rows in a result set. The new **ROW_NUMBER** expression in DB2 UDB for iSeries V5R4 provides exactly that capability (as demonstrated in the example at the top of this chart).

If your application needs the ability to restart this arbitrary row number for a subset or subgrouping of the results, the **ROW_NUMBER** expression has a partitioning clause that fills this need. You can see this in the second example where the row numbering restarts each time a new department number (that is, the partition) is processed.

OLAP expressions (RANK and DENSE_RANK)

- **RANK** and **DENSE_RANK** for highlighting a data attribute (independent of the result set sorting)

```
SELECT empno, lastname, salary+bonus AS TOTAL_SALARY,
       RANK() OVER (ORDER BY salary+bonus DESC) AS Salary_Rank
FROM employee WHERE salary+bonus > 30000 ORDER BY lastname
```

EMPNO	LASTNAME	TOTAL_SALARY	SALARY_RANK
000050	GEYER	40975.00	5
000010	HAAS	53750.00	1
200010	HEMMINGER	47500.00	2
000090	HENDERSON	30350.00	11
200220	JOHN	30440.00	9
000030	KWAN	39050.00	6
000110	LUCCHESI	47400.00	3
000220	LUTZ	30440.00	9
000070	PULASKI	36870.00	7
000060	STERN	32750.00	8
000020	THOMPSON	42050.00	4

SALARY_RANK
5
1
2
10
9
6
3
9
7
8
4

OLAP expressions (RANK and DENSE_RANK)

When ordering the results of a query, it is often necessary to apply a ranking to highlight another data attribute. For example, you might need to get a returned list of sales people that is ordered by their last names, but which also includes their ranking in terms of total revenue. In the example on this chart, the programmer uses the **RANK OLAP** expression to return a list that sorts the employees by last name and then ranks them according to the total salary.

There is actually a closely related ranking expression known as **DENSE_RANK**. The **RANK** and **DENSE_RANK** expressions behave in an almost identical fashion except when the values being ranked contain duplicate values. If you examine the results of the **RANK** output shown in the graphic, you will see that:

- Employees John and Lutz earned the same salary and shared a ranking of nine.
- Because two people occupied the rank of 9. The next employee got a ranking of 11.
- In other words, no one received a salary rank of 10.

If you look at the output of the **DENSE_RANK** statement for the same query on the right:

- Employees John and Lutz again share a ranking of nine.
- But, the next employee in the salary ranking has a ranking of 10 instead of 11.
- There is no gap in the salary ranking for this dense rank.

Recursive SQL

▪ Recursive common table expressions

- Useful for navigating tables where rows are inherently related to other rows in same table (bills of material, organizational hierarchies, and so forth)

– Example:

```
WITH emp_list (level, empid, name) AS
  (SELECT 1, empid, name FROM emp
   WHERE name = 'Carfino'
  UNION ALL
   SELECT o.level + 1, next_layer.empid, next_layer.name
   FROM emp as next_layer, emp_list o
   WHERE o.empid = next_layer.mgrid )
SELECT level, name FROM emp_list
```

1 – Initializing the query

2 – Recursive reference to the next level

3 – Start the query and return final results

The diagram illustrates the three phases of a Recursive Common Table Expression (RCTE) using color coding and arrows. The SQL code is shown with three distinct color groups: green for the initialization phase, blue for the recursive phase, and red for the final results phase. Arrows point from descriptive text to the corresponding parts of the code.

- Phase 1 (Green):** The initial query: `(SELECT 1, empid, name FROM emp WHERE name = 'Carfino')`. An arrow points from "1 – Initializing the query" to this section.
- Phase 2 (Blue):** The recursive reference: `UNION ALL SELECT o.level + 1, next_layer.empid, next_layer.name FROM emp as next_layer, emp_list o WHERE o.empid = next_layer.mgrid)`. An arrow points from "2 – Recursive reference to the next level" to this section.
- Phase 3 (Red):** The final results: `SELECT level, name FROM emp_list`. An arrow points from "3 – Start the query and return final results" to this section.

Recursive SQL

DB2 UDB for iSeries has supported common table expressions since OS/400 V4R4. With i5/OS V5R4, common table expressions have become even a more useful and powerful SQL feature with the ability to perform recursive processing. A common table expression that includes a recursive reference is known as a recursive common table expression (RCTE). Database tables can contain rows that have an inherent relationship to other rows in the same table; recursive algorithms support natural navigation through these tables. Examples of these types of tables include organizational hierarchies, bills of material, and travel connections.

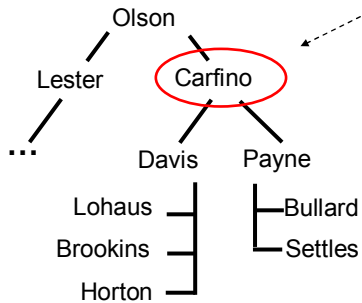
From the recursive example shown on this chart, the color coding and arrows show you that there are three phases in an RCTE. These phases are:

- Initializing the query (shown in green)
- Recursively navigating to the next level (shown in blue)
- Kicking off the recursion and returning the final results (shown in red)

The next several charts walk through each of these phases in more detail and explain the capabilities of recursive expressions.

Recursive SQL example (initialization phase)

**SELECT 1, empid, name FROM emp
WHERE name = 'Carfino'**



**CREATE TABLE emp(
empid INTEGER PRIMARY KEY,
name VARCHAR(10),
salary DECIMAL(9, 2),
mgrid INTEGER)**

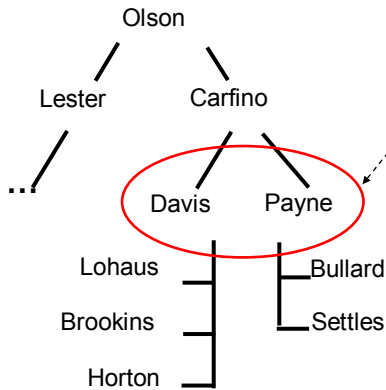
Recursive SQL example (initialization phase)

In this RCTE example, the initialization phase helps identify all the employees who work directly for a manager named Carfino. Thus, the first step is to retrieve Carfino's employee ID. The next phase involves finding all the employees who have a manager ID value that references Carfino's employee ID.

A constant value of 1 is returned on this seeding **SELECT** statement (shown in green) so that the final report can provide the organizational level or depth of an employee in Carfino's organization chain. The DB2 UDB engine will run the initialization statement only once; that is, it will not execute the initialization statement on any recursive executions of the **emp_list** table expression. However, DB2 UDB for iSeries requires you to code the initialization statement(s) at the beginning of the RCTE.

Recursive SQL example (recursion pass #1)

```
SELECT o.level + 1, next_layer.empid, next_layer.name
FROM emp as next_layer, emp_list o
WHERE o.empid = next_layer.mgrid
```



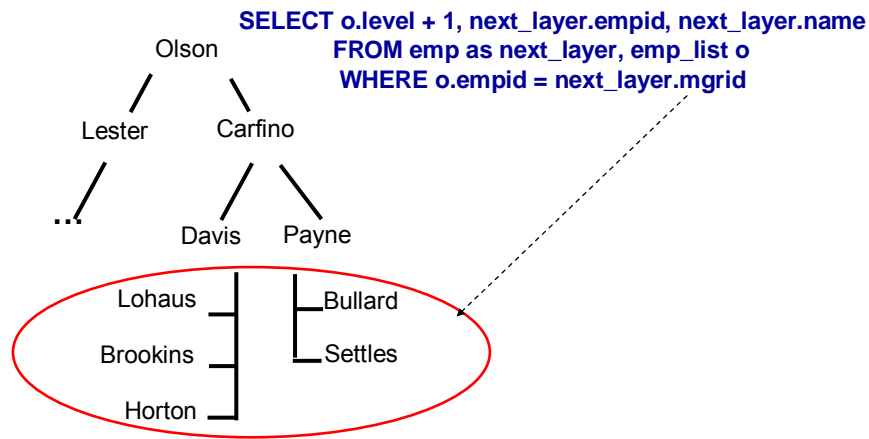
```
CREATE TABLE emp(
  empid INTEGER PRIMARY KEY,
  name VARCHAR(10),
  salary DECIMAL(9, 2),
  mgrid INTEGER)
```

Recursive SQL example (recursion pass #1)

The blue **SELECT** statement is the recursive portion of the RCTE because it contains a recursive reference to the **emp_list** table expression on the **FROM** clause.

The first time through the recursive phase of the query, the **SELECT** statement will produce a list of the employees that have Carfino as a manager. Davis and Payne will appear on this list because the **o.empid** column for these two employees will contain Carfino's employee number. (Carfino's **o.empid** value is joined and compared to the manager ID for all employees.) The join column on the recursive **SELECT** statement always needs to join to the seed value produced by the initialization statement (for example, the **empid** column). This join condition is the method by which the RCTE feeds itself with a list of employees to investigate on the next recursive execution. Also, notice that, on each execution, the **SELECT** statement increments the level column to track the recursion depth within the organization.

Recursive SQL example (recursion pass #2)



```

CREATE TABLE emp(
  empid INTEGER PRIMARY KEY,
  name VARCHAR(10),
  salary DECIMAL(9, 2),
  mgrid INTEGER)

```

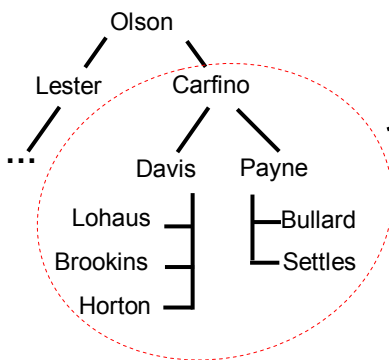
Recursive SQL example (recursion pass #2)

The last execution of this recursive expression will then produce a list of the employees that have Carfino's direct reports (Davis and Payne) as managers. The recursive traversals of the RCTE stop when the statement encounters employees who are not managers (meaning that the **mgrid** column never references their employee numbers).

Recursive SQL example (final results)

```

WITH emp_list (level, empid, name) AS
  (SELECT 1, empid, name FROM emp
   WHERE name = 'Carfino'
  UNION ALL
   SELECT o.level + 1, next_layer.empid, next_layer.name
   FROM emp as next_layer, emp_list o
   WHERE o.empid = next_layer.mgrid )
  SELECT level, name FROM emp_list
  
```



00001	NAME
1	Carfino
2	Payne
2	Davis
3	Settles
3	Bullard
3	Lohaus
3	Horton
3	Brookins

Recursive SQL example (final results)

The main **SELECT** statement (in red) starts the recursive processing of the organizational hierarchy and is a requirement to actually run the RCTE and produce a result set.

This **SELECT** statement references the **emp_list** common-table expression. This reference causes the seeding **SELECT** statement to execute first. Then, control passes to the recursive phase, which then constructs the final result set (displayed on this chart). The **UNION ALL** operation combines the results from the initialization and recursion phases.

The final result set does not list employees under their respective managers. By default, the DB2 UDB engine will always search and process the data in breadth-first order (by traversing across one layer of the organization before going on to the next layer).

Recursive SQL considerations

- **Breadth First** is default processing order
 - Use the **SEARCH** clause to change processing order (as long as the **Order By** clause is also specified).
 - Use the **Search** clause only as required (adds extra overhead).

```
WITH emp_list(level,empid, name) AS
  (SELECT 1,empid, name FROM emp
   WHERE name = 'Carfino'
  UNION ALL
   SELECT level+1, next_layer.empid, next_layer.name
   FROM emp as next_layer, emp_list p
   WHERE p.empid = next_layer.mgrid)
SEARCH DEPTH FIRST BY empid SET seqcol
SELECT level,name FROM emp_list ORDER BY seqcol
```

Depth search output



00001	NAME
1	Carfino
2	Davis
3	Brookins
3	Lohaus
3	Horton
2	Payne
3	Bullard
3	Settles

Recursive SQL considerations

DB2 UDB for iSeries V5R4 allows you to do a depth-first search through the support it provides for the RCTE. This is important because applications often need to view the data in this order. In a depth-first search, when the DB2 UDB engine encounters one of Carfino's direct reports, it processes all the employees who work for that manager before processing another one of Carfino's direct reports. The programmer specifies this processing order with the **SEARCH** clause (demonstrated in the query and results contained on this chart). It is obvious that the order of the depth-first result set on this chart naturally matches the relationships in an organization chart.

The first thing to notice in this depth-first example is that the **SEARCH** clause needs to use the same column that the recursive join reference specifies (in this case, **empid**). In addition to this navigation column, the **SEARCH** clause also contains a **SET** keyword with another column name, **SeqCol**. The DB2 UDB engine uses this column to keep track of the order in which it visited the rows on this SQL request. The sequencing column specified by the **SET** column is important because you must specify this same column name on the **ORDER BY** clause. The column name itself is not important; therefore, you are free to choose any name.

If the **ORDER BY** clause does not reference this sequencing column, then the DB2 UDB engine ignores the entire **SEARCH** clause. If you want your data returned in a specific order, then an **ORDER BY** clause is the only way to guarantee that ordering.

Recursive SQL considerations (continued)

- If table data has cyclic relationships, need use **CYCLIC** clause to prevent never ending queries

```
WITH destinations (departure, arrival, itinerary) AS
(SELECT f.departure, f.arrival, CAST (f.departure ||'-'|| f.arrival AS VARCHAR(200))
 FROM flights f
 WHERE f.departure ='Rochester'
 UNION ALL
 SELECT r.departure, b.arrival, CAST (r.itinerary ||'-'|| b.arrival AS VARCHAR(200))
 FROM destinations r, flights b
 WHERE r.arrival =b.departure )
CYCLE arrival SET cyclic_data TO '1' DEFAULT '0'
SELECT departure, arrival, itinerary, cyclic_data FROM destinations
```



DEPARTURE	ARRIVAL	00003	CYCLIC_DATA
Rochester	Chicago	Rochester->Chicago	0
Rochester	New York	Rochester->Chicago->New York	0
Rochester	Seattle	Rochester->Chicago->Seattle	0
Rochester	Minneapolis	Rochester->Chicago->Minneapolis	0
Rochester	Rochester	Rochester->Chicago->Minneapolis->Rochester	0
Rochester	Detroit	Rochester->Chicago->Minneapolis->Detroit	0
Rochester	Chicago	Rochester->Chicago->Minneapolis->Rochester->Chicago	1

Recursive SQL considerations (continued)

Another part of a recursive common table expression is the **CYCLE** clause. In some cases, your tables might contain recursive data relationships that are cyclic or never-ending. In an organization hierarchy, you do not have to worry about getting into a never-ending cycle because you do not have many company presidents that report to entry-level employees at the bottom of the organization. On the other hand, consider a table such as an airline **Flights** table that contains the air flight combinations for a set of cities.

Just after the **CYCLE** keyword on this example query is the name of the column that the DB2 UDB engine must watch to ensure that no RCTE enters a never-ending cycle. On the **SET** keyword, you supply a column so that the DB2 UDB engine can flag cyclic paths that it finds during execution of the RCTE. In this example, the DB2 UDB engine sets the **cyclic_data** column to the string value of **0** for noncyclic results and a value of **1** for any cyclic paths found. The output of this recursive SQL request also demonstrates that you can include the **cyclic_data** column in the final result set.

In the result set for this example, the DB2 UDB engine has marked the last row in the result set as cyclic and has ended the processing. When the **Chicago** destination showed up a second time in this set of flight connections, the recursive processing stopped. The DB2 UDB engine detected this unending cycle because the column specified on the **CYCLE** clause (arrival) caused the database engine to ensure that no value in the arrival column was processed twice on a single set of flight connections.

SELECT advancements

▪ Subquery and scalar fullselect

```
UPDATE MyExchangeRates m SET conversion_rate=
  (SELECT rate FROM EuropeRates e WHERE e.countryid=m.country_id
   UNION
   SELECT rate FROM AsiaRates a WHERE a.countrycode=m.country_id)/100
```

- Scalar fullselect is allowed anywhere an expression is supported (for example, Col1+50)
 - CASE, CAST, BETWEEN predicate, column function, and so forth

▪ Multicolumn predicates

- ...WHERE (c1,c2) IN (SELECT a,b FROM...)
- ...WHERE (c1,c2) = (SELECT a,b FROM...)

▪ Exclusive lock option (use with caution)

```
SELECT * FROM orders
WHERE order_id = 'E5100'
WITH RS USE AND KEEP EXCLUSIVE LOCKS
```

SELECT advancements

The iSeries platform completed compliance with the SQL standard with the addition of an SQL standard syntax **flagger** to the SQL flagger facility, as well as support for the subquery and scalar **fullselect** options. The **Fullselect** option enables multiple **SELECT** statements combined together with a **Union**, **Intersect**, or **Except SET** operator. Even though DB2 UDB for iSeries has supported the **fullselect** option for many releases, subqueries or scalar **subselect** options did not allow them.

The example **UPDATE** statement (shown on this chart) demonstrates the new scalar **fullselect** option support. In this example, the company has a table with currency exchange rates for all the countries with which it does business. On a daily basis, the company receives updated rates in tables from two sources. Exchange rates for European countries come from one source and rates for Asian countries come from another. DB2 UDB for iSeries V5R4 simplifies applications because a single SQL **UPDATE** statement can refresh the rates by combining the sources with the **UNION** operator. In prior releases, this same rate refresh task required two separate **UPDATE** statements.

In addition to the **fullselect** option support, subqueries are also easier to code because of the addition of multicolumn predicate support. Notice in the two predicate example how multiple columns can be compared with the **IN** and **=** predicates instead of having to code separate subqueries for each column being compared.

The new **KEEP EXCLUSIVE LOCKS** clause allows an application to acquire exclusive locks (instead of read locks) on rows that meet the selection criteria. This can be useful if the application reads a row that it intends to change, and you do not want to take the risk of another application locking that same row for update while the application processes the newly read data in preparation for updating that row. You must use the **KEEP EXCLUSIVE LOCKS** clause with caution because the exclusive row locks, which are acquired, will also prevent concurrent access by other users to those rows until the end of your transaction.

Richer toolbox of SQL functions

- **Triple DES data encryption (ENCRYPT_TDES)**
 - Will be available across the DB2 UDB family
 - Data encryption white paper
ibm.com/servers/enable/site/education/abstracts/4682_abs.html
- **Date and time processing**
 - LAST_DAY, NEXT_DAY, ADD_MONTHS, VARCHAR_FORMAT
- **Statistical processing**
 - STDDEV_SAMP and VARIANCE_SAMP
- **Miscellaneous**
 - GENERATE_UNIQUE
 - TABLE_NAME
 - RAISE_ERROR

```
SELECT emp_name,
CASE job_type
WHEN 1 THEN 'Programmer'
WHEN 2 THEN 'Administrator'
WHEN 3 THEN 'Project Manager'
WHEN 4 THEN 'Manager'
ELSE RAISE_ERROR('70001', 'Invalid JobType') END
FROM employee
```

Richer toolbox of SQL functions

A richer and deeper set of SQL built-in functions also simplifies life for SQL programmers who are trying to address new business requirements.

Encrypting data that is stored in DB2 UDB for iSeries is another common business problem. In this release, a new built-in encryption function, **ENCRYPT_TDES**, complements the **ENCRYPT_RC2** function that was delivered in the last release. (Refer to the **Column Encryption in IBM DB2 UDB for iSeries** course in the **Resources** section for more details on this topic). This new encryption function provides SQL developers with the ability to secure sensitive data with the Triple Data Encryption Standard (DES) algorithm.

Other new SQL functions help with the processing of date and time values. These include the **Last_Day**, **Next_Day**, **Varchar_Format**, and **AddMonths** functions.

Not surprisingly, you can use the new **GENERATE_UNIQUE** SQL function to return a unique value in the form of a bit data character string that is 13 bytes long. Each successive value is greater than the value returned on any previous execution of the function. This can be useful when the application requires a unique identifier for internal processing and does not need that value to be understandable to an end user.

The **Raise_Error** function gives you the ability to signal an exception from an SQL statement based on a specific calculation or condition. You might use this to alert the application that the query has detected an unexpected event. For example, the query shown at the bottom of this chart will terminate and signal **SQLSTATE 70001** if it encounters an invalid **job_type** value during execution of the **SELECT** statement.

There are also two new statistical functions to provide standard deviation and variance calculations that are based on the sampling of a data set, instead of processing every row in a table or result set.

INSTEAD OF triggers

- **New trigger type changes semantics of INSERTs, UPDATEs, and DELETE operations against a view (must be defined over an SQL view)**

- Many views are read-only because of derivations, joins, grouping, and so forth.
- **INSTEAD OF** triggers are useful in setting up encryption to be semi-transparent.
- New support builds on base **INSTEAD OF** trigger support delivered via i5/OS V5R3 PTF (ibm.com/iseries/db2/iot.html) by providing support for join views, and others.

```
CREATE VIEW empdept AS
```

```
  SELECT empno, firstname, lastname, deptname FROM employee, department
  WHERE workdept=deptno
```

```
CREATE TRIGGER UpdateJoin
```

```
  INSTEAD OF UPDATE ON empdept
```

```
  REFERENCING OLD ROW AS o NEW ROW AS n
```

```
  FOR EACH ROW
```

```
  BEGIN
```

```
    UPDATE employee
```

```
      SET empno=n.empno, firstname=n.firstname, lastname=n.lastname
```

```
      WHERE empno=o.empno;
```

```
    UPDATE department SET deptname = n.deptname WHERE deptname=o.deptname;
```

```
  END
```

INSTEAD OF triggers

The **INSTEAD OF** trigger is a new type of trigger for SQL views that DB2 UDB for iSeries V5R3 actually introduced via PTF. That support (delivered mid-release) had a number of restrictions that DB2 UDB for iSeries V5R4 eliminates. Unlike the existing trigger support, you must create **INSTEAD OF** triggers over an SQL view. They can never reference a table or physical file.

This chart contains an example of creating an **INSTEAD OF** trigger over an SQL view that contains a join. (The restrictive DB2 UDB for iSeries V5R3 support did not allow this.) This example demonstrates how you can use **INSTEAD OF** triggers to provide updateable join support on the **empdept** view.

Whenever the DB2 UDB engine updates the **empdept** SQL view, the **INSTEAD OF** trigger (**UpdateJoin**) allows the developer to ensure that the update succeeds. Without this trigger defined, the DB2 UDB engine would return an error that the view is read-only and cause the **UPDATE** request to fail. The **UpdateJoin** trigger transforms the original **UPDATE** request into two **UPDATE** statements against the two underlying tables (employee and department) that are joined together in the **empdept** view.

Miscellaneous SQL and DB2 enhancements

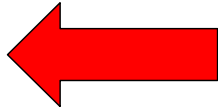
- **ANS timestamp format ('2005-06-13 01:22:03.000444')**
- **Support for LABEL ON INDEX**
- **Optional journaling for SQL tables (NOT LOGGED INITIALLY)**
`CREATE TABLE new_materials`
`AS (SELECT * FROM materials WHERE YEAR(item_date)=2006)`
`WITH DATA NOT LOGGED INITIALLY`
- **Simpler SQL and command line integration with new qcmdexc stored procedure**
`CALL qcmdexc('ADDLIBL DBLIB2', 15)`
instead of `CALL qcmdexc ('ADDLIBL DBLIB2', 0000000015.00000)`
- **Syntax flagger for SQL core standard**

Miscellaneous SQL and DB2 UDB enhancements

Here are some miscellaneous enhancements:

- Support for the ANS Timestamp format will be a welcome addition for SQL developers (especially those who port applications to the System i5 platform). In past releases, programmers had to change to a different format.
- The NOT LOGGED INITIALLY clause is a new option that is available on the CREATE and ALTER TABLE statements.
 - This new clause provides an SQL interface to end journaling temporarily. (For example, if you need to populate a work table with some data to get started and there is no requirement to commit or rollback the new data, you can use the **NOT LOGGED** clause to speed performance during this initial data population by temporarily ending journaling. At the end of the transaction, journaling automatically restarts on the table.)
- Some SQL applications might need to interact with the i5/OS operating environment using system command line (CL) operations. That interaction is simpler in DB2 UDB for iSeries V5R4 with an improved **QCMDExc** stored procedure that is easier to code for CL integration.

Agenda

- **Application development enhancements**
- **SQL and DB2 UDB enhancements**
- **Performance enhancements** 
- **Availability and recovery enhancements**
- **Ease-of-use and management improvements**

Performance improvements

Performance is critical to today's applications, even though the requirements for almost any application effort continue to become more complex. You will be pleased with some of the strong performance announcements with i5/OS V5R4 and DB2 UDB for iSeries.

Performance enhancements

- **SQL Query Engine (SQE) enhancements**
 - Support for LIKE, LOB columns, and sensitive cursors
 - Enhanced partitioned table optimization
 - Autonomic indexes
- **Faster SQL procedural language with expression evaluator**
- **Faster DB2 XML Extenders**
- **Transaction manager enhancements**
 - Parallel rollback unlock
 - Soft commit (reduce disk forcing with QIBM_TN_COMMIT_DURABLE environment variable)
- **Accelerator for System i5 platform boosts performance for SQL-intensive workloads on low-end i5-i520 servers as well as Java and WebSphere workloads**

Performance enhancements

The SQL Query Engine (SQE) is one of the key initiatives for taking SQL performance to new levels on the System i5 family. DB2 UDB for iSeries V5R4 SQL workloads will benefit from the query engine's new ability to process SQL requests that have a **LIKE** clause and that process large object (LOB) columns.

A more exciting SQE enhancement is the extra intelligence in the SQE query optimizer that improves performance with the automatic creation of autonomic indexes. The optimizer sits in the background and looks for frequently executed queries that can benefit from the creation of an index. If the optimizer identifies such a query, then DB2 UDB for iSeries creates an autonomic index as a temporary index and maintains it as the underlying table data changes. (This is similar to a permanent SQL index.) The autonomic index, however, will disappear when the system shuts down or restarts.

If you make heavy use of commitment control, you will also be able to recover faster with the parallel rollback option and the ability to customize commitment-control disk behavior to achieve better performance.

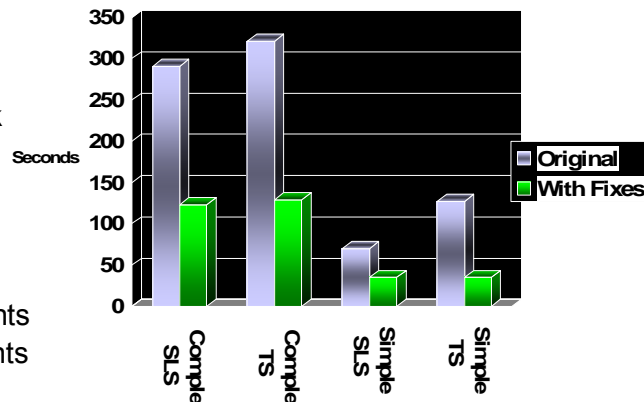
Significant DB2 XML Extender enhancements

Faster XML processing

- Performance boost of 50-90% from enhanced middleware stack (C runtime, XML parser, XML Extender engine)

Enhanced functionality

- Ability to compose XML documents that contain same named elements
- Ability to shred documents containing wrapper elements
- Increased scalability
 - Maximum size of XML document is 100 MB (up from 15 MB).
 - XML Extenders can now insert up to 512,000 rows into a table.



	i5/OS V5R3	i5/OS V5R4
Required PTFs	SI21015 SI21062	SI21069 SI20203 SI21062

➤ **New functionality and most of the performance enhancements are also available on i5/OS V5R3!**

Significant XML extender enhancements

Enhancements in the operating system and in the DB2 XML Extenders code have shown between 50% and 90% improvement in performance on some XML-based workloads.

In addition, enhancements allow you to use XML Extenders with more complex types of XML documents. These advancements include shredding (or decomposition) capabilities for wrapper elements and the ability to generate XML documents with shared element names.

SQE requested SQL enhancements

- **The following SQL enhancements are only available to the programmer if the SQL request is executed by SQE (otherwise SQL0255 is returned).**
 - Recursive CTE
 - RowNumber, Rank, and Dense_Rank
 - Subquery and scalar fullselect
 - INTERSECT and EXCEPT set operators
 - MQT optimizer awareness
 - iSeries Navigator SQL plan cache tool
- **Remaining SQE restrictions**
 - Sort sequences
 - User defined table functions
 - Logical file references
 - Select and omit logical files
 - Non-SQL interface
 - Read triggers
- **DB2 UDB for iSeries V5R4 includes a statement of direction for sort sequence support of RCTE, RowNumber, Rank, Subquery, and scalar fullSelect in future release.**

SQL enhancements available only to SQE

Notice that the DB2 UDB for iSeries V5R4 SQL features listed on the top of this chart are only supported by the SQL query engine. This means that, if your environment or SQL request contains an attribute that SQE does not support and you are trying to use one of these new SQL features, then DB2 UDB for iSeries will flag an error. Check your application and system settings carefully before deciding to use one of these new SQL enhancements.

Lack of support for national language sort sequences is probably the biggest SQE restriction (especially for systems in the European and Asia-Pacific geographies). As you can see on the bottom of this chart, IBM has issued an official statement of direction. In the next release of DB2 UDB for iSeries, sort sequences will no longer prevent the use of SQE with the new SQL technology that this release introduces.

Enhanced performance tooling

- **Predictive temporary storage governor**
 - STORAGE_LIMIT QAQQINI option
 - QRYSTGLMT parameter on CHGQRYA command
- **Governor exit point for easier administration, QIBM_QQQ_QUERY_GOVR**
- **CURRENT DEGREE special register for controlling DB2 SMP parallelism via SQL**
- **VOLATILE table support**
`CREATE TABLE worktable (id INT, name CHAR(10), current_total DEC(8,2)) VOLATILE`
- **New filters on STRDBMON command**
 - JOB wildcarding (QZDAS*)
 - RUNTHLD, FTRFILE, FTRUSER, FTRINTNETA
- **New IBM Redbook on database monitor**
ibm.com/redbooks
Diagnosing SQL Performance on DB2 UDB for iSeries (SG24-6654)

Enhanced performance tooling

Since IBM OS/400 V3R1, DB2 UDB for iSeries has featured a predictive query governor that is based on execution time (with the **CHGQRYA** (Change Query Attributes) command). If the query optimizer estimates that a query will take longer than the defined time limit, then an administrator can prevent the long-running query from taking over the system for an extended period of time. This is important when users execute ad-hoc queries.

Ad-hoc queries cause another issue; if the administrator has not properly tuned the database with the right indexes, then the query optimizer might have to create temporary objects (such as hash tables) to help in the execution of the query. If the server has lots of memory and temporary storage, the system can survive an ad-hoc query that uses a large number of temporary objects during its execution. However, if there are several concurrently running ad-hoc queries with this personality, it will not be long before the help desk phone is ringing off the hook with users complaining about response time issues. Thus, in DB2 UDB for iSeries V5R4, you can use a temporary storage governor via the **STORAGE_LIMIT QAQQINI** option or the Query Storage Limit parameter on the **CHGQRYA** command to provide another control for ad-hoc queries that tie up system resources. Just as with the time limit, the query optimizer will estimate how much space will be consumed by the temporary objects it needs to use during the query execution. If the estimated storage exceeds the user-defined limit, then you can prevent the query from running.

To better automate workflow when exceeding the governor storage or time limit, a new exit point is available in DB2 UDB for iSeries V5R4 for the query governor. You can use the **QIBM_QQQ_QUERY_GOVR** exit point to call a program every time the governor detects that a limit has been exceeded. This program can then notify an administrator or keep track of the users who are hitting the limits for future consultation.

DB2 UDB for iSeries: V5R4 Overview

Volatile tables are not a new type of SQL object. Volatile table support involves an attribute that you can specify when creating a normal SQL table. This new attribute indicates to the DB2 UDB query optimizer that the number of rows at runtime can vary widely and tells the optimizer to use an access method that performs relatively well (regardless of the table size). Many times, this will bias the query optimizer towards performing an index scan over this table when a query references it. Good candidates for this new keyword include work tables that are repeatedly populated and then cleared during a batch process.

Advanced database monitor filtering is one of the key attributes of the graphical DB2 On Demand Performance Center, which is part of iSeries Navigator. You will learn more about this graphical set of performance tools later. On this chart, you see the new parameters for the Start Database Monitor (**STRDBMON**) system command to enable more intelligent filtering of the monitor data that is collected. This CL command is the underlying technology behind the database monitor advancements included in the DB2 On Demand Performance Center.

DB2 UDB family and MQTs (stage 2 i5/OS V5R4)

- **MQT example:**

```
CREATE TABLE Example_MQT AS
  (SELECT Geography, Region, Year, Month,
    SUM(Revenue) AS Total_Revenue, SUM(Quantity) AS Total_Quantity,
    COUNT(*) AS Rows_per_Group
  FROM Example_Table
  GROUP BY Geography, Region, Year, Month)
DATA INITIALLY IMMEDIATE
REFRESH DEFERRED
ENABLE QUERY OPTIMIZATION
MAINTAINED BY USER
```
- **Optimizer can use the MQT instead of fully executing the following query:**

```
SELECT Geography, Year,
  SUM(Revenue) AS Total_Revenue, SUM(Quantity) AS Total_Quantity,
  FROM Example_Table WHERE Year IN (2004, 2005)
  GROUP BY Geography, Year;
```

White paper: ibm.com/servers/enable/site/education/abstracts/438a_abs.html
- **User responsible for keeping MQT data current and activating optimizer MQT awareness with QAQQINI options**

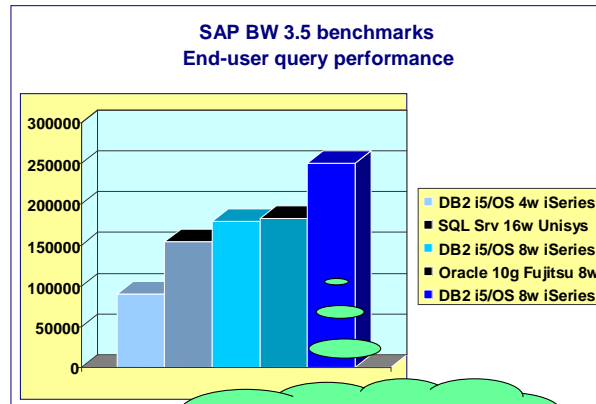
DB2 UDB family and MQTs (stage 2 i5/OS V5R4)

Materialized query tables (MQTs) are another feature that only SQE can use. Just as with **INSTEAD OF** triggers, a DB2 UDB for iSeries V5R3 PTF introduced MQTs, and this latest release enhances those capabilities. An MQT is a DB2 UDB table that contains the results of a query, along with the query's definition. Similar to autonomic indexes, SQE can use them to improve query performance automatically. When an SQL query runs, the SQE query optimizer determines whether any MQTs on the system match (or partially match) the query. When the query optimizer does find a match (or partial match), it will rewrite the query to use the contents of the MQT as a short cut to speed up the delivery of the query results. MQTs are most beneficial in data warehousing and business intelligence environments because the data does not change as frequently as in a transactional system.

System i5 platform optimized for SAP



- DB2 UDB for iSeries optimization
 - Specific features designed for SAP
 - Joint development since 1999
- System i5 Solution Edition
 - Preloaded hardware and software SAP appliance
 - Best selling Solution Edition
- Data warehouse benchmark leadership
 - SAP Business Warehouse benchmark certifications*
 - 12 months of leadership benchmarks
 - Several configurations tested
 - Superior scaling per processor
- Continued TCO leadership for SAP
 - IT costs for iSeries scenarios averaged 45.5% less than those for Sun Fire and 42.0% less than for Integrity**



**DB2 UDB for iSeries
regains leadership on
January 23, 2006**

*See detailed certified benchmark results at sap.com/solutions/benchmark/BW2_results.htm

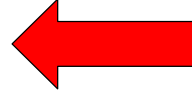
** ITG Study of SAP Installations, November 14, 2005. Comparisons are based on six composite profiles of manufacturing companies with revenues between \$500 million and \$12 billion. Input from 20 companies was employed to construct profiles.

System i5 platform optimized for SAP

Just as with the other DB2 UDB products, the i5/OS operating system and DB2 UDB for iSeries include many enhancements that specifically optimize the execution of SAP solutions running on the System i5 platform. MQT support in DB2 UDB for iSeries was one of these enhancements. The SAP Business Information Warehouse benchmark results highlighted on this chart demonstrate the performance benefits that MQTs can deliver on data warehouse report and query workloads.

Agenda

- **Application development enhancements**
- **SQL and DB2 UDB enhancements**
- **Performance enhancements**
- **Availability and recovery enhancements**
- **Ease of use and management improvements**
- **Tools and utilities**



Availability and recovery enhancements

High availability goes hand-in-hand with performance as an enterprise-level concern. In this section, you will learn about more robust catalogs and cross-reference files, as well as journaling enhancements.

More robust catalogs and cross-reference files

- Queue protection
- RCLSTG SELECT(*DBXREF) progress indicator
- Library-level reclaim with RCLDBXREF command
- Automatic rebuild QSYS2 and SYSIBM catalog views when recreating cross-reference files

More robust catalogs and cross-reference files

To improve availability and on-demand capabilities, DB2 UDB for iSeries V5R4 improves the robustness of the cross-reference files that support the catalog views. The improved protection of the queue responsible for populating these files reduces the number of times that there is a need for recovery efforts. In the rare cases that a recovery of the cross-reference files is still necessary, the new i5/OS **RCLDBXREF** command allows for a more granular recovery at a library level, and system-wide recoveries now include a progress and status indicator. In addition, DB2 UDB for iSeries will also automatically rebuild the catalog views when necessary to eliminate the need for manual intervention.

Journaling enhancements

- **Improved automatic journal configuration**
 - Support for **SEQUENCE** object
 - Automatic SQL journaling with **QDFTJRN** data area after **CrtDupObj** and restore commands
 - Data Area layout...**
 - Bytes 1-10: Journal Library**
 - Bytes 11-20: Journal Name**
 - Bytes 21-30: Object Type (*ALL,*FILE,*NONE, *DTAARA, *DTAQ, *RSTOVRJRN)**
 - Bytes 31-40: Option (*ALLOPR,*CREATE, *MOVE, *RESTORE)**
- **SMAPP and journaling support for encoded vector indexing (EVIs) and ICU indexes**
- **Ability to journal minimal data journal entries with *FLDBDY option**
- **Enhanced remote journal error reporting**
- **Customizable journal recovery Count**
- **Enhanced CHGJRN and WRKJN commands**
- **Single journal support for 10,000,000 objects**
- **New F-IT entry for identity columns**

Journaling enhancements

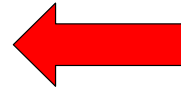
Journaling (or logging) is a key component of any business solution that needs to be highly available. The i5/OS journaling capabilities have improved greatly with the last couple of releases, and i5/OS V5R4 is no exception. There are enhancements to automatic journaling for SQL-created databases, as shown on this screen.

Journal management is also simpler with enhancements to the Change Journal (CHGJRN) and Work Journal (WRKJRN) system commands. There is also a new maximum object limit.

The journal minimal data feature was introduced a couple of releases ago to reduce the size of journal objects on the system by writing only the data that actually changed to the journal. The new Field Boundary (***FLDBDY**) option still minimizes the amount of data written to the journal, but it does it at a field level instead of a byte level. (This allows the minimized journal entries to be viewable for audit purposes.)

Agenda

- **Application development enhancements**
- **SQL and DB2 UDB enhancements**
- **Performance enhancements**
- **Availability and recovery enhancements**
- **Ease-of-use and management improvements**

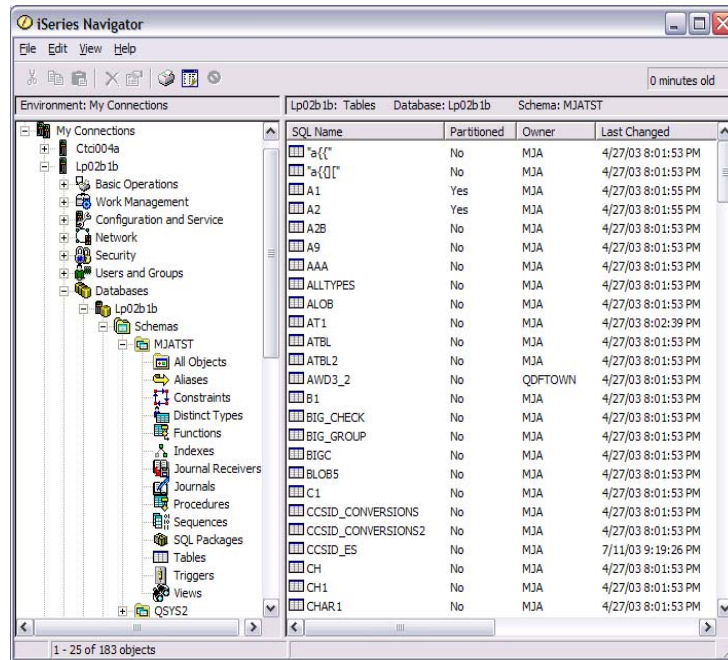


Ease-of-use and management improvements

The final section of this course reviews the following:

- iSeries Navigator enhancements
- Improved index advice
- Live DB2 UDB performance analysis
- Advanced database monitor filtering
- Simplified database monitor analysis
- Other management enhancements (including DB2 Content Manager Standard Edition V8.3)

iSeries Navigator enhancements



- **DB2 UDB On Demand Performance Center**
- **Health Center**
- **Import and export**
- **Table actions**
 - CLRPFM command
 - CPYF command
 - INZPFM command
- **EDTRBDAP support**
- **Show current SQL**
 - Statement name
 - Program or package name
 - Open information
- **Delimiter option on creates**

iSeries Navigator enhancements

As mentioned earlier, the enhancements to iSeries Navigator V5R4 include some of the most notable DB2 UDB improvements. Ignoring the DB2 On Demand Performance Center and Health Center, most of the enhancements listed on this chart are minor. With many of these enhancements, the only enhancement was the addition of a graphical interface for system commands (such as the Copy File **[CPYF]** command).

The next few screens cover the more significant enhancements to this System i5 graphical interface.

Enhanced index advice

System-wide Index Advisor (always active)

Table for Which Index was Advised	Schema	Keys Advised	Index Type Advised	Last Advised for Query Use	Times Advised for Query Use
ADDRESSES	DB2ADM	LINK_ID,LOCATION,TYPE	Binary Radix	7/6/05 5:22:53 AM	82
HISTORICAL_VARIATION_PER...	COORSO16	ENVIRONMENT	Binary Radix	9/20/05 10:22:30 AM	82
T_02G2TMD05	DATAWFD	TIPOLOGIA_ID, TIENDA_ID	Binary Radix	12/31/05 5:55:27 AM	82
T_02G2TMD02	DATAWFD	TIPOLOGIA_ID, TIENDA_ID	Binary Radix	12/31/05 5:55:27 AM	82
T01061MD04	DATAWFD	SECCION_ID	Binary Radix	1/1/06 5:20:42 AM	82
T01061MD03	DATAWFD	SECCION_REF_ID	Binary Radix	1/1/06 5:20:42 AM	82
BPRTOPL_DATA_PERIOD	DB2ADM	TOPL_PERIOD,TOPL_YEAR,TOPL_STOCKROOM_CODE	Binary Radix	7/6/05 8:05:24 AM	81
CUST_DIM	STAR100G	CUSTKEY	Encoded vector (not unique)	12/29/05 12:09:47 AM	80
FEMEQ	CORPDB	JDE_EQUIP_TAG	Encoded vector (not unique)	12/28/05 11:31:52 PM	80

Index advice, also enhanced (SQE only)

- SQE optimizer will also examine join, grouping, and ordering criteria
- Available on i5/OS V5R3 with latest database group PTF

Enhanced index advice

The database component of iSeries Navigator is the major driver of simplification in DB2 UDB for iSeries V5R4 with the addition of new tools and the streamlining of existing tools. This new combination of tools is known as the DB2 On Demand Performance Center.

The system-wide Index Advisor is one of the new tools in i5/OS V5R4. Index advice from the DB2 UDB query optimizer is not a new capability in DB2 UDB for iSeries V5R4. In past releases, you might find feedback on suggested indexes in the optimizer debug messages within a job log or a database monitor collection. However, this feedback was only available if someone manually started a database monitor or turned on debug messaging. If analysts did this, they had to deal with the complexities of extracting the advised index details of various joblogs and database monitor files. The new Index Advisor eliminates these hurdles with the click of a mouse. There is no need for user intervention, such as starting a database monitor, because the query optimizer automatically logs index advice for all queries to a repository in DB2 UDB for iSeries V5R4. Just right-click the database name in the iSeries Navigator tree and select the **Index Advisor** task to get the output displayed on this screen capture.

Even better is the fact that the index advice provided in DB2 UDB for iSeries V5R4 by the SQE query optimizer is more intelligent and complete. The index advice given in past releases focused only on the query's filtering criteria and ignored the join, grouping, and ordering criteria. Again, this design required manual intervention from an analyst to take the DB2 UDB index advice and then review the associated query to supplement it. Look closely at the Index Advisor output and you will see that the optimizer also provides advice on the type of index to create. Some queries might benefit from a traditional radix index, while others will benefit from an encoded vector index structure.

Live DB2 UDB performance analysis via SQL plan cache

SQL Plan Cache Statements - Tplxe3(Tplxe3)

Filters for statement list:

- ☐ Minimum runtime for the longest execution: 0 Seconds
- ☐ Queries run after this date and time: Aug 25, 2005 4:19:36 PM
- ☐ Top 'n' most frequently run queries: 0
- ☐ Top 'n' queries with the largest total accumulated runtime: 0
- ☐ Queries ever run by user:
- ☐ Queries currently active
- ☐ Queries with index advised
- ☐ Queries with statistics advised
- ☐ Include queries initiated by the operating system
- ☐ Queries that use or reference these objects:

Schema Table Browse

☐ SQL statement contains:

Clear Selections Retrieve

List of statements:

Last Time Run	Most Expensive Time (sec)	Total Processing Time (sec)	Total Times Run	Statement	User
Aug 25, 2005 11:26:07 AM	1008.8919	1008.8919	1	INSERT ...	TATO
Aug 25, 2005 4:15:33 PM	0.2588	2.3101	461	UPDATE ...	QSYS
Aug 25, 2005 4:04:40 PM	1.0043	1.7914	14	SELECT ...	CMWAI
Aug 25, 2005 4:04:45 PM	0.5374	1.3219	14	SELECT ...	CMWAI
Aug 25, 2005 4:04:55 PM	0.6137	1.2407	14	SELECT ...	CMWAI
Aug 25, 2005 4:04:50 PM	0.4763	1.1449	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:00 PM	0.4433	0.8920	14	SELECT ...	CMWAI
Aug 25, 2005 4:04:36 PM	0.4390	0.7912	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:17 PM	0.3672	0.6612	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:59 PM	0.2040	0.6236	14	selectte...	CMWAI
Aug 25, 2005 4:06:26 PM	0.2395	0.5843	14	selectte...	CMWAI
Aug 25, 2005 4:06:13 PM	0.3105	0.5637	14	selectte...	CMWAI
Aug 25, 2005 4:06:53 PM	0.2364	0.4478	14	selectte...	CMWAI
Aug 25, 2005 4:05:29 PM	0.2374	0.4364	14	SELECT ...	CMWAI
Aug 25, 2005 4:06:40 PM	0.2193	0.4259	14	selectte...	CMWAI
Aug 25, 2005 4:05:26 PM	0.1464	0.3945	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:33 PM	0.1261	0.3300	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:07 PM	0.1399	0.2691	14	SELECT ...	CMWAI
Aug 25, 2005 7:50:24 AM	0.1040	0.2065	69	SELECT ...	QDIRS
Aug 25, 2005 4:07:06 PM	0.0424	0.1995	14	selectte...	CMWAI
Aug 25, 2005 4:05:22 PM	0.0767	0.1964	14	SELECT ...	CMWAI
Aug 25, 2005 4:05:00 PM	0.0032	0.1400	14	SELECT ...	CMWAI

Selected statement:

```
INSERT INTO DWLIBM.DWWSTVNM SELECT
DWWSTSKA.WEEKID ,
DWWSTSKA.STRID ,
DWSKUD.SKUVNDID ,
DWWSTSKA.CUSTTYPE ,
DWWSTSKA.SOFLAG ,
DWWSTSKA.EVENT ,
DWWSTSKA.SALTYP ,
SUM ( DWWSTSKA.SALGTY ) AS SSALQTY ,
SUM ( DWWSTSKA.SALAMT ) AS SSALAMT ,
```

Show Longest Runs Run Visual Explain

Create Snapshot Refresh

Close Help ?

Always on (no database monitor overhead)

Live DB2 UDB performance analysis via SQL plan cache

Another tool that is always up and running (like the Index Advisor) is the SQL plan cache toolset; the SQL plan cache is an internal repository that SQE uses to store the access plans and associated statistics for SQL statements that are currently running or have recently executed on the system. The big benefit with this plan cache analysis toolset is that, for the first time, you can perform detailed SQL performance analysis without the overhead of running a database monitor. A detailed database monitor collection can eat up disk space in a hurry and noticeably slow disk response times. Gone are the days when a user calls about poor performance and the only response for an SQL-based application was to start a database monitor and rerun the report or application. Now, when there is a performance issue on i5/OS V5R4, you can analyze the contents of the SQL plan cache to see if a long-running SQL statement is the cause of the performance problems. Using the SQL plan cache tool is as simple as right-clicking the **SQL Plan Cache Snapshots** component and selecting the **SQL Plan Cache > Show Statements** task. This action will produce the screen shown here.

One feature of the SQL plan cache viewer is the statement filters that are available on the left side of the dialog. These filters allow you to intelligently subset the SQL statements that you want to analyze, as opposed to scrolling through a random list of thousands of SQL statements. For instance, you can pick the top 10 statements that execute most often or take the most time. You can also choose to analyze the statements currently running or the SQL statements that a particular

DB2 UDB for iSeries: V5R4 Overview

user runs. Furthermore, when you have selected an SQL statement to analyze, you can seamlessly launch Visual Explain for that statement from the same window.

You will need to remember that the SQL plan cache is a fixed size. Therefore, some SQL statements might get purged out of the cache if they have not recently executed. In these situations, you will use the database monitor tool.

If you want to archive the contents of the cache into a permanent object, you can create an SQL plan cache snapshot. Effectively, this operation converts the contents of the SQL plan cache into a table that has a format similar to a detailed database monitor collection. After you create the snapshot, you can run reports to analyze the contents, just as you can with a database monitor collection.

Advanced database monitor filtering

SQL Performance Monitor Wizard - Tplxe3(Tplxe3)

To limit the amount of data collected, specify which filters to use. When filters are provided, only statements that match the specified filter values will be captured.

If you would like to limit the amount of data collected specify which filters to use:

☐ Minimum estimated query runtime:

☐ Job name:

☐ Job user:

☐ Current user:

☐ Internet address:

☒ Queries that access these tables:

Schema	Table Name

Browse... Remove

Activity to monitor

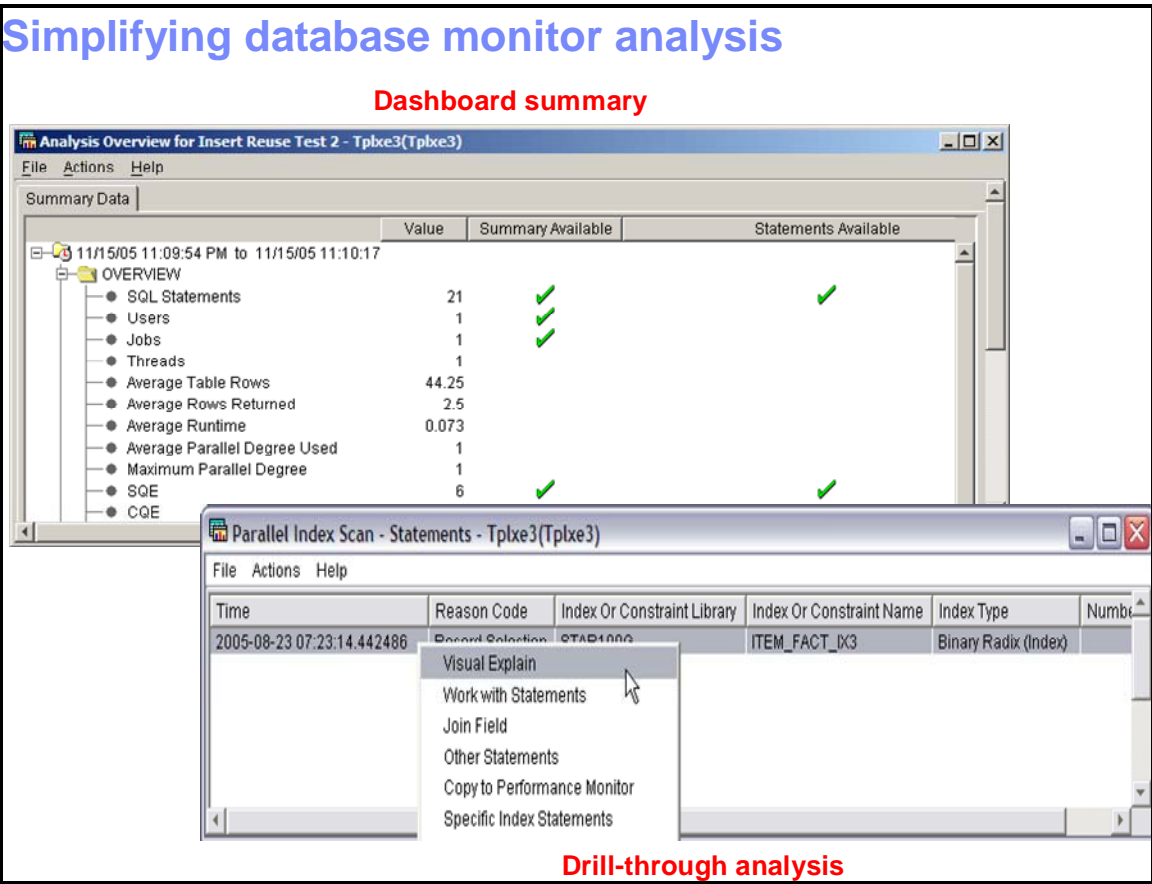
☒ Only collect monitor output for user activity

☐ Collect monitor output for user and system activity

Help ? < Back Next > Finish Cancel

Advanced database monitor filtering

Database monitors have existed on DB2 UDB for iSeries for many releases, but the usability of these monitors takes a big step forward in this release with addition of both pre- and post-filters. These filters are similar to the filters that were available with the SQL Plan Cache viewer. Database monitor filters are important for a couple of reasons. First, the filters can minimize the overhead and disk space that a database monitor collection consumes. As a case in point, an IBM customer recently ran a database monitor collection on all jobs on the system for 15 minutes. An SQL-based ERP application that was running on the system at the time resulted in almost three million rows of detailed monitor data. Applying one of the filters available in DB2 UDB for iSeries V5R4 (see screen capture), such as collecting monitor data only for a certain user or only for SQL statements that reference a specified table, can significantly reduce the amount of data collected. These filters are available when you start a new SQL Performance Monitor collection via the graphical interface or the **STRDBMON** CL operation.



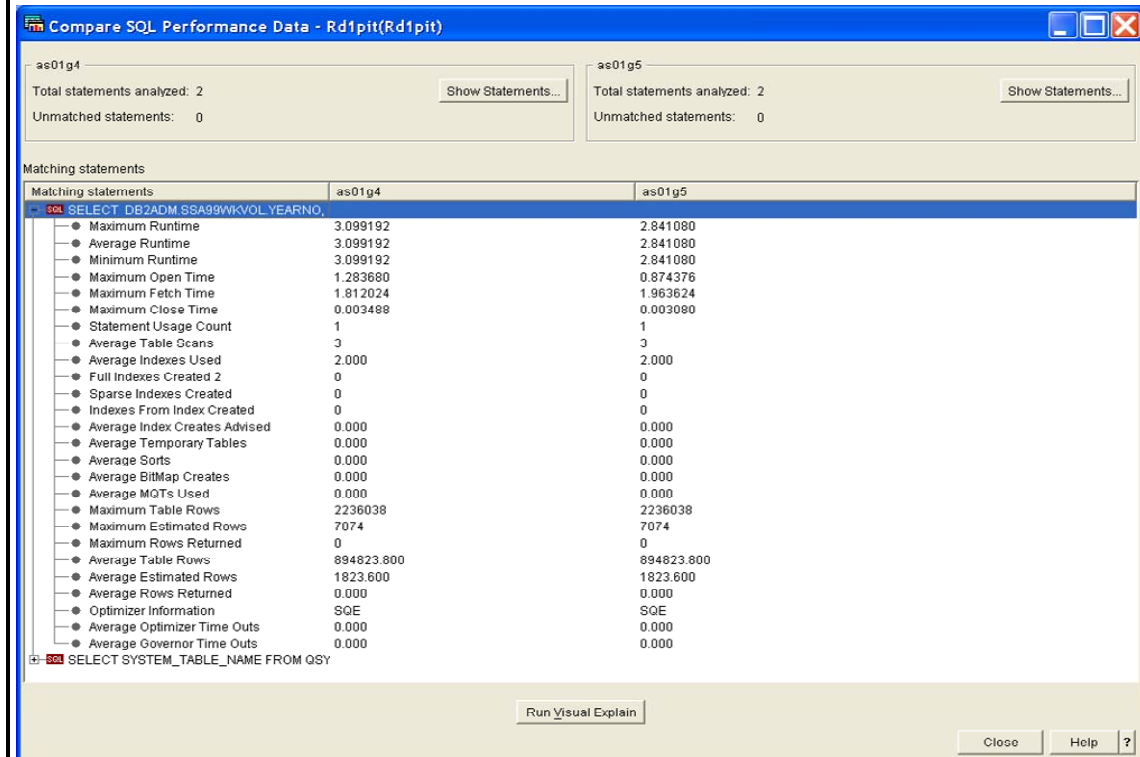
Simplifying database monitor analysis

The filters discussed on the previous chart eliminate the number of rows in a database monitor collection, but there is still the need to sort through hundreds of fields in a single database monitor row and many monitor row types in order to identify and fix DB2 UDB performance problems. This maze of data makes it a challenge to look at a database monitor collection quickly in order to determine if there are any issues (unless you are a DB2 UDB performance expert).

Thus, DB2 UDB for iSeries V5R4 includes a dashboard summary to address this issue. On the first analysis of a database monitor collection, the DB2 UDB engine will automatically summarize some of the key performance indicators in the collection and then present the results in the summary window depicted at the top of this screen capture. This summary will allow an analyst to scroll quickly through some high-level DB2 UDB for iSeries performance indicators to determine if there is a need for more detailed analyses.

DB2 UDB for iSeries V5R4 also addresses another issue with the database monitor analysis reports by providing drill-through analysis. Previously, these reports were helpful in finding an SQL statement with performance issues. However, for more detailed research of this statement (with a tool such as Visual Explain), you had to exit out of the tool and then manually copy that SQL statement for analysis. As you can see from the bottom figure, you can easily right-click to drill down into more detail with Visual Explain. This will be a great enhancement for improving the efficiency of database analysts and administrators.

Simplifying database monitor analysis (continued)



Monitor comparison utility

Simplifying database monitor analysis (continued)

Often, the performance of an application can change over time as the system configuration changes and new versions are installed. Unfortunately, the process of making changes and installing new software releases can sometimes have an unexpected negative impact on performance. With the new monitor comparison utility, you can utilize database monitor collections to help determine what change is causing performance issues. This technique requires that you gather and save database monitor collections when a critical DB2 UDB for iSeries workload or report is performing and running well on the system. If performance of this workload or report changes, you can compare a new database monitor collection against the good performance monitor collection to help identify the differences that are causing or contributing to the throughput problems.

This can be a long-running process with large database monitor collections; therefore you must plan accordingly.

Additional management enhancements

- **DB2 UDB Health Center**
 - Monitor database metrics
 - Check which objects are nearing DB2 UDB limits
- **Index evaluator (first available on i5/OS V5R3)**

SQL Name	Type	LAST QUERY USE	LAST QUERY STATISTICS USE	QUERY USE COUNT	QUERY STATISTICS USE
QASQRESL	LOGICAL FILE	2004-09-10 11:31:54	2004-09-10 11:31:54	586	746
Q_QSYS2_...	Primary Key Constraint	2004-09-09 15:51:19	2004-09-09 15:51:19	649	649

- **MQT evaluator**
- **OnDemand SQL analysis (enhanced Show Current SQL)**
 - Statement name
 - Program or package name
 - Open information

Additional management enhancements

One final iSeries Navigator enhancement to highlight is the DB2 UDB Health Center, which allows you to monitor and perform a checkup on how well your database objects are doing compared to different DB2 UDB for iSeries limits, such as the maximum number of allowable rows per table or the maximum size of an index object. In addition, you can change the thresholds that pertain to marking a DB2 UDB object as being close to a limit. You can also save these checkups to a permanent file to enable historical comparisons. You launch this feature by right-clicking your named database and selecting the **Health Center** task.

The index and MQT evaluators allow you to determine how valuable an index or MQT is by letting you check how often the DB2 UDB query optimizer uses it. This can be helpful for administrators and analysts who are trying to clean up unused DB2 UDB objects on the system. To access these evaluators, just right-click the table object and select the appropriate **Show** task.

If you want to determine if an SQL statement is contributing to a slow-moving application, the iSeries Navigator **Show Current SQL** tool is great to use. It has been available since DB2 UDB for iSeries V5R2, but with this latest release, the tool returns more attributes about the SQL request (such as program name) to help you more quickly identify the issue. You can access this tool by right-clicking the **Databases** icon in the iSeries Navigator tree.

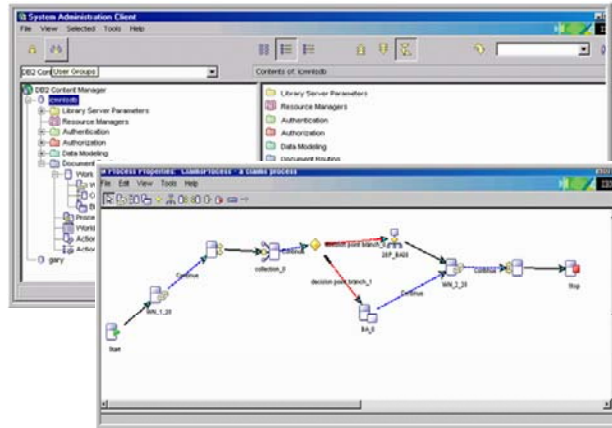
DB2 Content Manager Standard Edition V8.3

- Helps small to mid-sized businesses capture, integrate, manage, and deliver digital content (such as Web content, e-mail, documents, reports, and images)
- Intended for new DB2 UDB for iSeries customers (not to provide migration from DB2 Content Manager for iSeries V5.3)
- StorageView from MBS Technologies extends support for DR550 and many other storage devices

- Announcement 11/29/2005
- General availability 1/6/2006



For more information, see
ibm.com/software/data/cm/cmgr/mp/edition-standard.html



Storage options and DB2 Content Manager	Disk	Optical	DR550
DB2 Content Manager for iSeries			
V5.3	X	X	X (MBS)
Standard Edition V8.3	X	X (MBS)	X (MBS)
DB2 Content Manager OnDemand for iSeries			
Spool file archive	X	X	
Common server	X	X	Planned

DB2 Content Manager Standard Edition V8.3

IBM Content Management software captures, stores, manages, integrates, and delivers all forms of digital content across a company's value chain, from employees to customers to suppliers and partners.

DB2 Content Manager Standard Edition V8.3 provides an entry-level content management solution for small and medium-sized businesses. It also provides a single, open, comprehensive, and consistent framework to manage, share, reuse, and archive all types of digitized content. Content types include the following: HTML and XML Web content, document images, electronic office documents, and rich media.

DB2 Content Manager Standard Edition was initially available on the iSeries platform with DB2 UDB for iSeries V5R3. Its tools include the following:

- A Windows based eClient and administrative clients for desktop or browser access
- A single GUI interface to manage the system
- Servers to provide indexing, storage, retrieval, and archiving services to manage all types of digitized content and document routing
- Workflow options that support a formal, consistent, and repeatable document routing process

DB2 UDB for iSeries: V5R4 Overview

The Standard and Enterprise editions share a common code base and Windows type of management client. However, the Standard edition does not provide the full range of features for larger enterprises, such as hierarchical storage management (HSM), migration services (from DB2 Content Manager V5.3), full text search, and federated services (including advanced workflow).

DB2 Content Manager Standard Edition does not support HSM to optical or IBM System Storage™ DR550 devices. Therefore, it might be better to consider MBS® StorageView™ Storage Manager, which enables HMS support for System Storage DR550 and any other devices supported by IBM Tivoli® Storage Manager to DB2 UDB Content Manager Standard Edition (and DB2 Content Manager for iSeries V5.3).

Summary

Application flexibility and portability

- Free-format RPG and SQL
- Enhanced SQL standards support
 - Scalar fullselect
 - Standard flagger
- Improved DB2 UDB family compatibility
 - Recursive expressions
 - OLAP support (RowNumber and Rank)
 - ANS timestamp format
 - TDES encryption
 - 2-MB SQL statements and 32-KB keys

On-demand and availability

- Cross-reference files robustness
- Automatic journaling enhancements
- SMAPP for EVIs
- Parallel rollback unlock

Performance

- SQL query engine enhancements
 - LIKE and LOB support
 - Sensitive cursors
 - Autonomic indexes
- Faster SQL procedural language
- Enhanced MQT optimization
- Faster XML extenders

Usability

- iSeries Navigator enhancements
 - DB2 On Demand Performance Center
 - SQE plan cache analyzer
 - DB2 Health Center
- Resource governor
- Governor exit point
- DB2 Content Manager

Summary

DB2 UDB for iSeries V5R4 maintains its position as a leader in ease-of-use and integration. This latest release contains many new capabilities that continue to make it a database management system that is simple for programmers, analysts, and administrators to use.

iSeries Navigator (the i5/OS graphical interface) delivers the most simplification with the addition of the DB2 On Demand Performance Center that will dramatically simplify SQL performance analysis for database analysts and administrators. SQL programmers will be able to solve a wider range of business problems with additions such as recursive SQL, triple data encryption standard (DES) algorithm, and the integration of SQL with free-format RPG syntax.

DB2 UDB for iSeries, already proven to be a leader in data warehousing through recent benchmarks, enhances SQL capabilities to tackle the problems that affect businesses. With the addition of OLAP functions, recursive SQL, and improved materialized query table support, the database can now take on an even broader range of workloads.

DB2 UDB for iSeries V5R4 improves performance and scalability on a number of fronts. In particular, the SQL query engine (SQE) handles more DB2 UDB requests and now has autonomic index capability. SQL procedures, functions, and triggers will receive a significant performance boost with i5/OS V5R4.

There are some valuable additional resources listed in the **Resources** section of this course.

Resources

These Web sites supplement the information contained in this document:

- DB2 UDB for iSeries Web site
ibm.com/eserver/iseries/db2
- DB2 Connect Unlimited Edition for iSeries Web site
ibm.com/software/data/db2/db2connect/edition-uei.html
- DB2 Universal Database for iSeries articles and white papers
ibm.com/iseries/db2/awp.html
- IBM white paper: Column Encryption in IBM DB2 UDB for iSeries
ibm.com/servers/enablsite/education/ibo/view.html?oc#db2
- DB2 Universal Database Cursor Sensitivity and Query Optimization information
ibm.com/iseries/db2/sqe.html
- DB2 UDB Instead Of triggers
ibm.com/iseries/db2/iot.html
- Indexing and statistics strategies for DB2 UDB for iSeries education offering
ibm.com/server/enablsite/education/ibo/record.html?indxng
- DB2 Migration Toolkit download
ibm.com/servers/enablsite/db2/porting.html
- DB2 Query Management Facility Distributed Edition for Multiplatforms V8.1
ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=897&letternum=ENUS204-028
- DB2 Query Management Facility Web site with available evaluation version
ibm.com/qmf
- Available MQT support PTFs for DB2 UDB for iSeries V5R3
ibm.com/iseries/db2/mqt.html
- DB2 UDB for iSeries Quick-Start
ibm.com/iseries/db2/gettingstarted.html
- DB2 UDB for iSeries SQL Query Engine
ibm.com/iseries/db2/sqe.html

Newsgroups

- USENET: comp.sys.ibm.as400.misc, comp.databases.ibm-db2
- iSeries Network SQL and DB2 forum at:
www.iseriesnetwork.com/Forums/main.cfm?CFApp=59

Education resources (classroom and online)

- DB2 UDB for iSeries Quick Start
ibm.com/eserver/iseries/db2/gettingstarted.html
- DB2 UDB Education
ibm.com/eserver/iseries/db2/db2educ_m.html
- iSeries Developer Roadmap
ibm.com/eserver/iseries/roadmap
- DB2 UDB for iSeries online education offerings
ibm.com/servers/enablsite/education/ibo/view.html?oc#db2
- DB2 UDB performance workshop
ibm.com/eserver/iseries/service/igs/db2performance.html

White papers

- DB2 UDB for iSeries and Web Services
ibm.com/servers/enablsite/education/abstracts/db2web_abs.html

DB2 UDB for iSeries: V5R4 Overview

- The creation and use of materialized query tables within IBM DB2 UDB for iSeries
ibm.com/servers/enable/site/education/abstracts/438a_abs.html
- Indexing and Statistics Strategies for DB2 UDB for iSeries
ibm.com/servers/enable/site/education/abstracts/indxng_abs.html
- Column Encryption in IBM DB2 UDB for iSeries
ibm.com/servers/enable/site/education/abstracts/4682_abs.html
- General DB2 UDB white papers:
ibm.com/servers/enable/site/education/ibo/view.html?wp#db2

Self-study tutorials

- Piloting DB2 UDB for iSeries using iSeries Navigator in V5R3
ibm.com/servers/enable/site/education/abstracts/2dc2_abs.html
- Master SQL Performance Visual Explain
IBM will add when loaded to the Web-- Not connected at the moment
- Integrating XML and DB2 UDB for iSeries
ibm.com/servers/enable/site/education/abstracts/intxml_abs.html
- Check back for DB2 UDB for iSeries V5R4 tutorials at:
ibm.com/servers/enable/site/education/ibo/view.html?oc#db2

DB2 UDB for iSeries publications

- **Online manuals**
ibm.com/eserver/iseries/db2/books.htm
- **Porting help**
ibm.com/servers/enable/site/db2/porting.html
- DB2 UDB for iSeries Redbooks: ibm.com/redbooks
 - Stored Procedures, Triggers and User Defined Functions on DB2 Universal Database for iSeries (SG24-6503)
 - Preparing for and tuning the V5R2 SQL Query Engine on DB2 Universal Database for iSeries (SG24-6598)
 - Modernizing IBM eServer iSeries Application Data Access – A Roadmap Cornerstone (SG24-6393)
 - Integrating DB2 Universal Database for iSeries with Microsoft ADO .NET (SG24-6440)
- SQL/400 Developer's Guide by Paul Conte and Mike Cravitz (ISBN 0-07-212568-3)

DB2 UDB family certifications

- IBM Certified Database Associate – DB UDB V8.1 Family (Test 700)
ibm.com/certify/certs/dbdaudv81.shtml
 - Education resources:
ibm.com/certify/tests/edu700.shtml
 - DB2 Version 8 Family Fundamentals Preparation online tutorials:
www7b.boulder.ibm.com/dmdd/library/tutorials/db2cert/db2cert_V8_tut.html
- IBM Certified Application Developer - DB2 UDB V8.1 Family (Test 703)
ibm.com/certify/certs/dbapudv81.shtml
 - **Education resources:**
ibm.com/certify/tests/edu703.shtml
 - **Sample tests:** Refreshed and updated exams for DB2 UDB for iSeries
certify.torolab.ibm.com/ice

DB2 UDB for iSeries: V5R4 Overview

DB2 UDB miscellaneous resources

- Details on cursor sensitivity article, entitled “DB2 UDB for iSeries cursor sensitivity enhancements” at:
ibm.com/developerworks/db2/library/techarticle/dm-0403milligan
- Details on using the graphical debugger with stored procedures can be found in the white paper: *Graphical Debugger makes Procedural SQL Debug Even Easier* at:
ibm.com/servers/enable/site/education/abstracts/sqldebug_abs.html
- For additional information on IBM Rational, visit the IBM Rational support online center for software support and a quick reference guide at: ibm.com/software/rational/support/quick-ref-guide.html
- Free DB2 Migration Toolkit download
ibm.com/servers/enable/site/db2/porting.html
- *Using DB2 Development Center with DB2 UDB for iSeries* white paper
ibm.com/servers/enable/site/education/abstracts/db2dev_abs.html

Other resources

- See detailed SAP certified benchmark results at:
sap.com/solutions/benchmark/BW2_results.htm

Trademarks and special notes

© IBM Corporation 1994-2006. 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:

IBM	i5/OS	eServer	On Demand Business
The IBM logo	DB2	iSeries	Informix
ibm.com	DB2 Universal Database	POWER5	WebSphere
OS/390	DB2 Connect	System Storage	Redbooks
OS/400	System i5		

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.

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

Linux is a trademark of Linus Torvalds 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.

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

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.