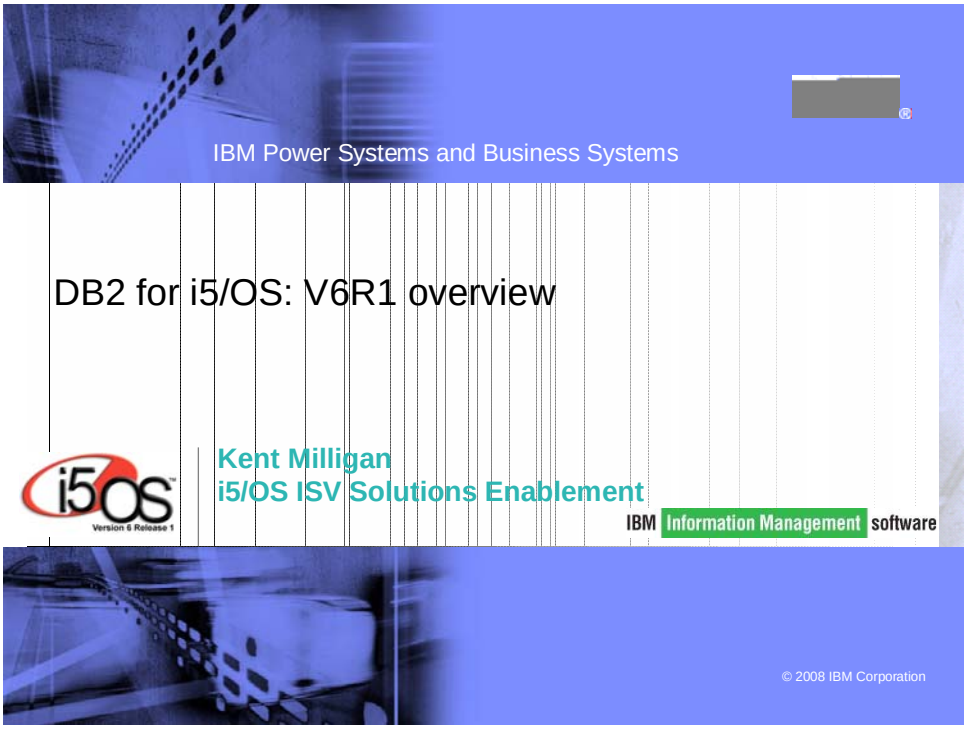
Table of contents

About the author

Kent Milligan

Kent Milligan is a DB2 technology specialist in IBM ISV Business Strategy and Solutions Enablement for the System i platform. After graduating from the University of Iowa, Kent spent the first eight years of his IBM career as a member of the DB2 development group in Rochester, Minnesota. He speaks and writes regularly on various DB2 for i5/OS relational database topics.



Introduction

Welcome to this online course, entitled “DB2 for i5/OS V6R1 overview.”

With the delivery of the latest release of the IBM® i5/OS® operating system, Version 6 Release 1, the highly popular and award-winning database system, IBM DB2® for i5/OS, is more powerful than ever. It offers more application flexibility and portability. It provides more integration between SQL and RPG, and it honors more Java™ and encryption standards. It is easier to use, easier to tune and supports easier Internet access options.

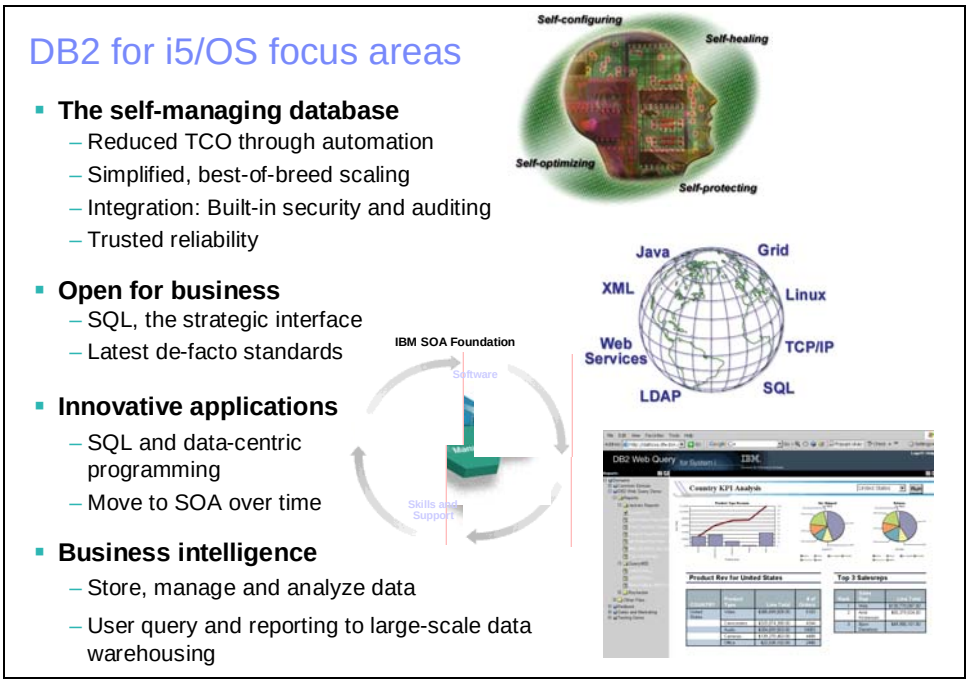
This course is important for all DB2 application developers who want their applications to run on the IBM System i™ platform. It is equally important for i5/OS developers who want to ensure that their applications are prepared to take advantage of the many new and valuable functions provided by DB2 for i5/OS V6R1.

Agenda

- **Enhancement overview**
- **Application-development enhancements**
- **SQL and DB2 enhancements**
- **Availability and recovery enhancements**
- **Performance enhancements**
- **Ease-of-use and management enhancements**

Agenda

In this course, you will first see a brief overview of the many enhancements that DB2 for i5/OS V6R1 delivers. Then, the course provides more detail about these enhancements, as shown on this chart.



DB2 for i5/OS focus areas

This chart shows the strategic directions for DB2 for i5/OS. As you progress through this course, you can relate the i5/OS V6R1 enhancements back to these strategic initiatives. The integrated relational database on the System i platform and its predecessor systems (IBM AS/400®) has always been known for its ease of use and simplicity. The tight integration with the i5/OS operating system allows DB2 to automatically handle many lower-level tasks (such as storage allocation) that require a database administrator (DBA) on other platforms. IBM plans to add even more self-managing capabilities to DB2 for i5/OS in future releases. Security, reliability and scalability are other long-time strengths that IBM continues to extend and leverage.

The System i platform and its built-in database, DB2 for i5/OS, support a wide variety of industry and de-facto standards to make accessing your business data as easy as possible. SQL is the strategic database interface for DB2 for i5/OS. Rich SQL support not only makes it easier for software vendors to port their applications and tools to the System i platform, but it also enables i5/OS developers to use industry-standard SQL for their data access and programming needs. The IBM DB2 Family shares this focus on SQL standards with DB2 for i5/OS, so this investment in SQL also enables DB2 for i5/OS to continue to leverage the relational-database technology leadership position of IBM and maintains close compatibility with the other DB2 Family products.

Despite heavy emphasis on SQL interfaces, DB2 for i5/OS continues to allow existing applications to run with no changes. This balancing act allows i5/OS developers to evolve and move to newer application technologies, such as SOA and data-centric programming, at their own pace.

IBM is also focusing on enhancing the data-analysis experience. Companies worldwide rely on DB2 for i5/OS to store mission-critical business data. Recent enhancements, such as IBM DB2 Web Query, make it much easier to analyze DB2 for i5/OS data. The continued IBM investment in business intelligence for the System i platform, along with this system's legendary security and reliability, help reduce costs and complexity, helping to eliminate the need to move data-analysis workloads to other platforms.

DB2 for i5/OS V6R1 enhancements

Application flexibility and portability
- SQL and RPG integration
- Enhanced Java Database Connectivity (JDBC) and Microsoft® .NET support
- SKIP LOCKED DATA clause
- Extended-indicator variables
- VALUES on FROM
- Hidden-timestamp columns
- Improved DB2 Family compatibility
  - Online analytical processing (OLAP) support – Cube and Rollup
  - INSERT on FROM
  - Unsupported-syntax tolerance
  - Advanced encryption standard (AES) encryption

On demand and availability
- Enhanced, online reorganization
- Library-level journaling

Performance
- SQL query engine enhancements
  - Sort sequence support
  - Self-learning optimizer
  - (Encoded-vector index) EVI-only processing
- Derived SQL indexes
- Faster Full Opens
- Client special registers

Usability
- System i Navigator enhancements
  - Customizable performance analysis
  - Spreadsheet integration
  - Plan cache enhancements
  - Index Advisor improvements
- IBM DB2 Web Query for System i

DB2 for i5/OS V6R1 enhancements

Keeping ahead of the competition is a top priority in today's ever-changing business environment. From an IT and database perspective, this means that you need to be able access and present business data in new, insightful ways to your users, and you need to deliver these capabilities yesterday. As you will learn in this course, the DB2 for i5/OS V6R1 enhancements include several features that enable your databases to run faster and allow your programmers to deliver new solutions quicker and easier.

This chart contains a categorization of the DB2 for i5/OS V6R1 enhancements that are discussed later in this course.

Application-development enhancements

Application-development enhancements

Next, you will learn some details about the i5/OS V6R1 enhancements that are important to your DB2 application-development efforts.

Enhancements for i5/OS application development

- **Improved RPG and SQL integration**
  - IFS source-file support
  - Improved variable scoping
  - Enhanced LIKE
    - Provides support for variables based on SQLCA variables
    - For example, SQLSTATE
  - Improved IBM WebSphere® Development Studio Client integration for source-code error resolution
  - SQL support in IBM application-development tools
    - SQL syntax highlighting
    - Templates for SQL statements
    - Formatting
- **ILE COBOL SQL precompiler support for Unicode data**
- **Enhanced SQL scripting**
  - RUNSQLSTM support for IFS files
    - Larger and wider SQL scripts
  - Improved DB2 Qshell utility

Enhancements for i5/OS application development

The IBM Integrated Language Environment® (ILE) SQL precompilers were enhanced with the source stream-file capability in i5/OS V6R1. Furthermore, IBM continued the momentum of improved SQL and RPG integration from previous releases with additional enhancements to the ILE RPG SQL precompiler. One of the key improvement areas in i5/OS V6R1 is the precompiler's ability to scope variables at a procedure level. In addition, the ILE SQL precompilers were enhanced with source stream-file support. Several other enhancements to the RPG SQL precompiler are also listed on this chart. The SQL advancements made to the IBM application-development tools are covered in later charts.

Not to be overlooked, COBOL programmers also enjoy a benefit from the Unicode support that has been added to the ILE COBOL SQL precompiler.

The Run SQL Statements (RUNSQLSTM) command is a popular tool with i5/OS developers who need to run SQL scripts from a command-language (CL) program or command-line interface (CLI). With i5/OS V6R1, the RUNSQLSTM command can process and run SQL statements that are stored in a stream file — previously, you had to store the SQL in a source physical-file member. This new stream file support can assist developers in several ways. Most importantly, SQL statements no longer have to be limited to 80 characters in length (or fewer) when stored in a stream file. This 80-character limit for statements that are stored in a source-file member made SQL coding quite difficult and tedious for i5/OS developers. In addition, a stream file accommodates larger, more complex SQL scripts because they have a 1 TB size limit, which is much larger than a source-file member's 16 MB limit. Finally, this support lets you manage i5/OS SQL scripts through a stream-file-based, change-management system.

RPG SQL precompiler variable scoping

```
PSubProc1          B                    EXPORT
D                  PI
D OutArray         ds                   qualified dim(1000)
D  customer                       25A
D  region                         25A
/free
   exec sql DECLARE c1 CURSOR FOR  SELECT customer,region FROM cust_dim;
   exec sql OPEN c1;
   exec sql FETCH NEXT FROM c1 FOR 100 ROWS INTO :OutArray;
   exec sql CLOSE c1;
 return;
 /end-free
P                  E
PSubProc2          B                    EXPORT
D                  PI
D OutArray         ds                   qualified dim(1000)
D  part                           55A
D  mfgr                           25A
D  brand                          10A
/free
  exec sql DECLARE c2 CURSOR FOR SELECT part,mfgr,brand FROM part_dim;
  exec sql OPEN c2;
  exec sql FETCH NEXT FROM c2 FOR 200 ROWS INTO :OutArray;
  exec sql CLOSE c2;
  return;
 /end-free
```

i5/OS V5R4 PTFs are planned.

RPG SQL precompiler variable scoping

As mentioned, the ability of the RPG SQL precompiler to scope variables at a procedure level is one of the key additions in DB2 for i5/OS V6R1. This new capability allows the SQL precompiler to properly process and run code that is similar to the RPG code you see on this chart.

In this example, two RPG procedures use the same variable name, *outArray*, to declare data structures — but each procedure declares slightly different attributes. The first procedure declares the outArray data structure with two fields, and the second procedure contains three fields within the outArray declaration. In addition, each procedure contains an SQL FETCH statement that references the duplicated variable name. In prior DB2 releases, the SQL RPG precompiler was unable to scope the variable name to the procedure, which resulted in complications for RPG developers. The effect of this deficiency was that the SQL RPG precompiler either failed with an error that flagged the duplicate variable names as illegal (even though it is valid RPG syntax) or unpredictable results occurred at run time because the SQL RPG precompiler incorrectly shared one definition of the duplicated variable across multiple procedures. The i5/OS V6R1 SQL RPG precompiler removes this hurdle by correctly scoping the variables at a procedure level. In addition, IBM plans to provide a PTF in 2008 to make this variable-scoping enhancement also available on i5/OS V5R4.

Improved industry-standard application interfaces

- **JDBC**
  - JDBC 4.0
  - Alias support in metadata APIs
  - Returning DEFAULT
  - Return update counts
- **ADO.NET**
  - Exploitation of ADO.NET 2.0
  - Integration with Microsoft Visual Studio
  - Distributed transactions
  - Multiple-row INSERT
  - Enhanced data-type support
- **CLI**
  - Wide API support for Unicode data
  - Alias support in Metadata APIs
  - Row-wise array INSERT
  - Support for complete ISO timestamp

Improved industry-standard application interfaces

DB2 for i5/OS V6R1 improves Java application-based data access with the Java Database connectivity (JDBC) Version 4.0 enhancements, along with other miscellaneous improvements.

The Microsoft® .NET data provider in i5/OS V6R1 exploits version 2.0 of ADO.NET and provides support for distributed transactions and multiple-row INSERT. In addition, this .NET provider delivers tighter integration with the Microsoft Visual Studio development environment by adding support for Visual Studio database interfaces and wizards such as Server Explorer.

When porting applications to DB2 for i5/OS, you will frequently find that the application uses SQL CLI as the data-access programming interface. The portability of these SQL CLI applications also increases in DB2 for i5/OS V6R1 with the addition of support for the CLI Wide-Character APIs. The CLI Wide-Character functions are used in applications that support Unicode or double-byte data.

RPG and SQL integration: Syntax highlighting

- **SQL syntax highlighting for both free- and fixed-format ILE RPG**
  - WebSphere Development Studio Client 7.0 and IBM Rational Developer for System i 7.1

```
000100      /free
000200        exec sql create procedure median_result_set
000300        language sql dynamic result sets 1
000400          BEGIN
000500            case v_workdept when 'B01'
000600            -- comments
000700            then update department set  deptname = 'DATA ACCESS 2';
000800            else update department set  deptname = 'DATA ACCESS 3;
000900              end case;
001000           end;
001100      /end-free
```

```
001500     C/EXEC SQL
001600  11 C+  DECLARE C2 CURSOR FOR
001700     C+    SELECT EMPPROJACT.PROJNO, PROJNAME, COUNT(*),
001800     C+       SUM((DAYS(EMENDATE) - DAYS(EMSTDATE)) * EMPTIME *
001900     C+           DECIMAL((SALARY/:WRKDAY),8,2))
002000     C+    FROM CORPDATA/EMPPROJACT, CORPDATA/PROJECT, CORPDATA/EMPLOYEE
002100     C+    WHERE EMPPROJACT.PROJNO = PROJECT.PROJNO AND
002200     C+          EMPPROJACT.EMPNO = EMPLOYEE.EMPNO AND
002300     C+          PRENDATE > :RDATE
002400     C+    GROUP BY EMPPROJACT.PROJNO, PROJNAME
002500     C+    ORDER BY 1
002600     C/END-EXEC
```

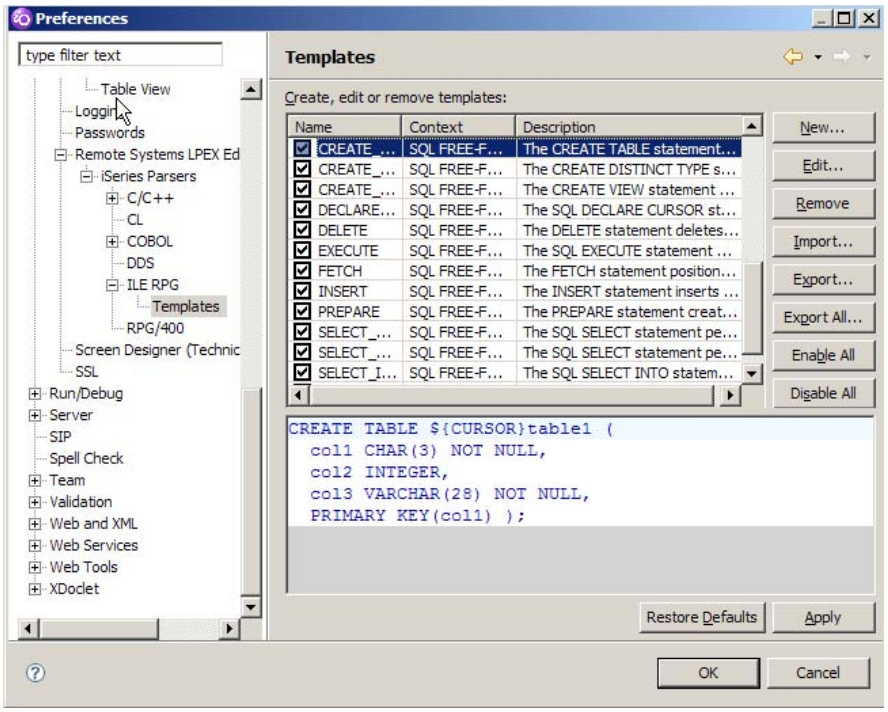
RPG and SQL integration: Syntax highlighting

Coding SQL statements into RPG programs is simpler with the new SQL syntax highlighting that is shown on this chart. The syntax highlighting is only available in IBM WebSphere® Development Studio Client 7.0 and IBM Rational® Developer for System i.

Syntax highlighting and syntax checking (which were added in Version 6.0.1 of these tools) does require that you properly register the SQL parser type for the source member type. You can do this by going to the tools' main page and clicking **Preferences -> Parser Association Settings**. Then, you can specify the ILErpgSQL parser value for any RPG and SQL source member types. You can also alter the various colors that are used for SQL syntax highlighting from the **Preferences** pull-down menu.

RPG and SQL integration: Templates

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



RPG and SQL integration: Templates

You can jumpstart your SQL coding efforts within RPG applications by using statement templates, which allow you to create an SQL statement shell that you can then automatically copy into your application source code from the Content Assist menu. These templates enable you to avoid typing in SQL keywords and, instead, just type in the user-defined parts of an SQL statement, such as the column or table name.

Reusability with extended-indicator variables

- **Make SQL statements more reusable**
  - Reuse single UPDATE statement (instead of coding an UPDATE statement for each distinct combination of columns)
  - Use indicator variables on INSERT VALUES and PREPARE statements

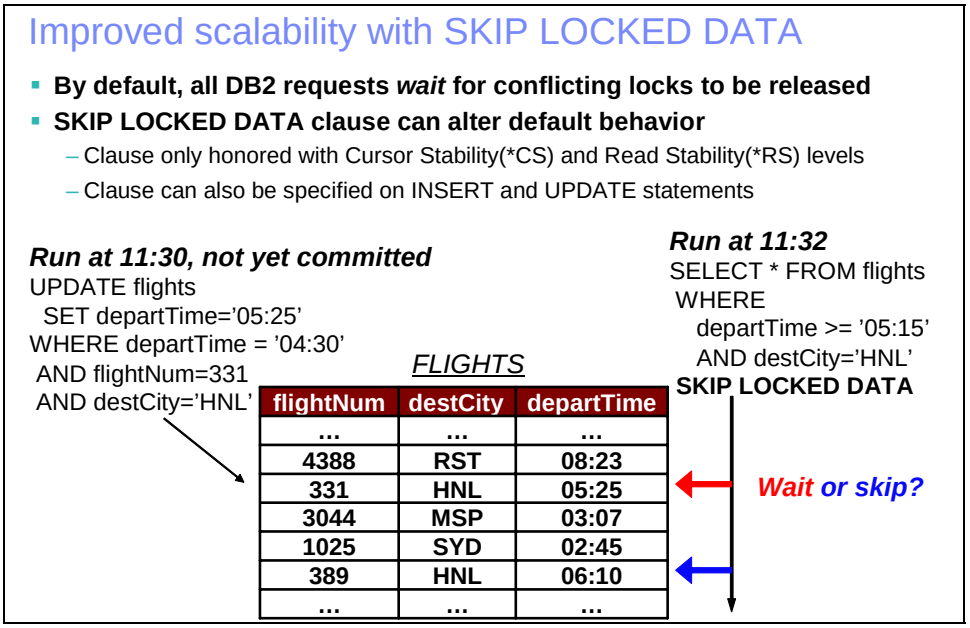
```
DECLARE cur1 CURSOR WITH EXTENDED INDICATORS FOR
      SELECT order_id, shipdate, quantity, status FROM orders;

OPEN cur1;
FETCH cur1 INTO :orddat:inds;
…
inds2 = -7;
UPDATE orders
      SET shipdate = :nsdate:inds1,
            quantity = :nqty:inds2,
            status = :nstat:inds3
      WHERE CURRENT OF cur1;
```

Indicator values	Meaning
0	Value is provided
-1,-2,-3,-4,-6	Null value
-5	Default value
-7	Column is ignored

Reusability with extended-indicator variables

Extended-indicator variables enable developers to code a single, generic UPDATE statement — instead of coding a different UPDATE statement for each combination of columns that need to be updated. The coding example on this chart shows this statement reusability. Notice the WITH EXTENDED INDICATORS clause on the cursor declaration. This clause allows extended-indicator variables to be passed on positioned UPDATE operations that are associated with the referenced cursor (that is, cur1). Extended indicators are then set to a defined set of special values to tell the UPDATE statement whether or not the column must be included in the update operation. In this example, the inds2 extended-indicator variable is set to a value of -7, which tells DB2 to omit the quantity column from the positioned UPDATE. Thus, the embedded UPDATE statement actually only changes two columns (shipdate and status), even though the SET clause references three columns. You can also use extended indicators with INSERT statements.



Improved scalability with SKIP LOCKED DATA

The SKIP LOCKED DATA clause might be the best addition to the SQL developer's toolbox in i5/OS V6R1. You can use this clause to increase scalability of applications that have heavy concurrent activity. There is one common misconception about DB2 locking: if you run a query against a DB2 table that other jobs and users are changing, DB2 automatically skips any row changes that are not yet committed. Actually, the opposite occurs — if a row matches the search criteria of the query; in that case, DB2 stops and waits to see if it can acquire the lock on that row.

To understand this behavior better, consider the following example, where the SQL requests run with the Cursor Stability (CS) isolation level:

> One night, you search the Internet for possible vacation flights to Honolulu. Two seconds before you submit your search on the travel agency's Web site, the agency's flight database performs an UPDATE on this chart to change the departure time from 4:30pm to 5:25pm. (This UPDATE is part of a large batch of updates to their flight schedules.)
>
> Your flight-search request issues the following SELECT statement to find a flight that leaves after 5:15 p.m. for Honolulu. When the SELECT statement processes the row that was just updated by the large batch-update process, it finds a match because the departure time was just updated to 5:25 p.m. and, therefore, meets the specified selection criteria.
>
> ```
> SELECT * FROM flights
>  WHERE departTime >= '05:15' AND destCity='HNL'
> ```
>
> At this point, DB2 does not know that the row has been locked, so it attempts to acquire a read lock on this row, as dictated by the CS isolation level. The UPDATE statement has not yet been committed because it is part of the batch update, so it holds an update lock on the row. Therefore, the update lock conflicts with the requested read lock, causing the SELECT statement to stop and wait until the row lock is released. If the batch-update process in this

example is long-running, the SELECT statement might fail with a lock-timeout error (the default record wait time is 60 seconds on i5/OS).

As mentioned, the fact that the SELECT statement stops and waits for the release of an update lock on the row is a surprise to many developers who believe that DB2 skips the locked row and searches for other rows in the table that meet the selection criteria. This might be good or bad, depending on the requirements of your application.

There is good news in i5/OS V6R1, you have a choice of behaviors. You can continue to run with the default behavior where DB2 waits for locks. Alternatively, you can have DB2 skip the locked rows. The SKIP LOCKED DATA clause activates the skipping behavior, as the SELECT statement on this chart shows.

With this clause specified, the SELECT statement skips any rows where it encounters a lock conflict, instead of waiting to acquire the row lock and possibly failing with a lock-timeout error. You can also specify the SKIP LOCKED DATA clause on UPDATE and DELETE statements. In addition, the clause is only honored with the CS and Read Stability (RS) isolation levels.

INSERT on FROM

- **Simplifies access to values that DB2 generates for identity columns**
  - Prior support (IDENTITY_VAL_LOCAL function) required overhead of an extra SQL statement
  - Prior support had no solution for blocked INSERT statements

Examples:

```
CREATE TABLE orders( order_id INTEGER AS IDENTITY,
                     order_date DATE,
                     order_qty INTEGER,
                     order_item CHAR(4));

SELECT order_id FROM FINAL TABLE (
   INSERT INTO orders VALUES(DEFAULT,'11/03/2007',50,'JM12'));

SELECT order_id FROM FINAL_TABLE (
   INSERT INTO orders VALUES(DEFAULT,'11/05/2007',12, 'JM09'),
                            (DEFAULT,'11/05/2007', 1, 'PC01'))
ORDER BY INPUT SEQUENCE;
```

Result 1

Result 2 3

INSERT on FROM

The ability to include an INSERT statement on a FROM clause is another enhancement that allows you to accomplish more processing on a single SQL statement. A number of i5/OS developers have taken advantage of the DB2 ability to automatically generate key values with the identity-column clause. One challenge with identity columns is accessing the key value (for example, order number) that DB2 generates — because it requires running an additional SQL statement. The INSERT statement is run to add the new row, followed by an extra SELECT statement with the Identity_Val_Local function to retrieve the generated value. This two-statement solution is also problematic when you insert a block of rows with a blocked INSERT, because the Identity_Val_Local function only returns the generated-key value for the last row inserted.

DB2 for i5/OS addressed these challenges in i5/OS V6R1 by enhancing the SELECT statement so that you can specify an INSERT statement on the FROM clause, as shown on the first SELECT statement on this chart. With this syntax, the INSERT statement on the FROM clause runs first and, then, the outer SELECT statement returns the values that DB2 generated for the order_id identity column. Notice that the FINAL TABLE keyword is required when referencing an INSERT statement on the FROM clause.

The second INSERT on FROM example demonstrates how you can use this new support to access the generated-identity values for a blocked-insert operation. The ORDER BY INPUT SEQUENCE clause allows you to get the generated values back for the order_id column in the order that DB2 generated them. Not only does this new INSERT on FROM syntax solve the issue of retrieving generated values for blocked inserts, but it also lets your application run slightly faster by allowing a single DB2 request to perform multiple database operations.

VALUES on FROM

- **Dynamically generate and populate temporary tables as part of the query definition**
  - Enable SQL access of in-memory tables that are maintained by application
  - Supports table-less queries

Examples:

```
SELECT deptnum, deptname FROM org WHERE deptnum <20
  UNION ALL
SELECT * FROM (VALUES(77,'New Department')) AS tmp(c1,c2)


WITH proposedRates(prType, prRate) AS ( VALUES(?,?),(?,?),(?,?),(?,?) )
SELECT rmtype, prRate, ((prRate – rmRate)/rmRate)*100 AS RateChgPercent
   FROM  rooms, proposedRates WHERE rmtype = prType
ORDER BY RateChgPercent DESC
```

VALUES on FROM

The FROM clause has also been improved with the ability to specify a VALUES clause. This support enables you to dynamically declare, populate and reference a temporary table on a SELECT statement as the examples on this chart show.

This new VALUES support is quite helpful for applications that use in-memory tables or data sets — where it is preferable that the application-maintained data is included and referenced by SQL statements. VALUES on FROM clauses also make it possible to include default or new values in a result set without the overhead of creating a real temporary-table object.

SQL and DB2 enhancements

SQL and DB2 enhancements

Next, you will learn some details about the i5/OS V6R1 enhancements to SQL that can be important to your DB2 application development efforts.

Data-access advancements

- **SQL OLAP extensions – GROUPING SETS and Super Groups**
- **More flexible FROM**
  - VALUES on FROM
  - INSERT on FROM
  - FULL OUTER JOIN
- **Expanded SQL Function Toolset**
  - Data encryption: ENCRYPT_AES
  - String processing: ASCII and CHR
  - Date and time processing
    - TIMESTAMP_FORMAT and VARCHAR_FORMAT
    - MONTHS_BETWEEN
    - ROUND_TIMESTAMP and TRUNC_TIMESTAMP
- **Limit improvements**
  - 120 columns on GROUP BY
  - 128-byte cursor and statement names
  - Partial support for 64 KB result-set width (i5/OS V5R4)

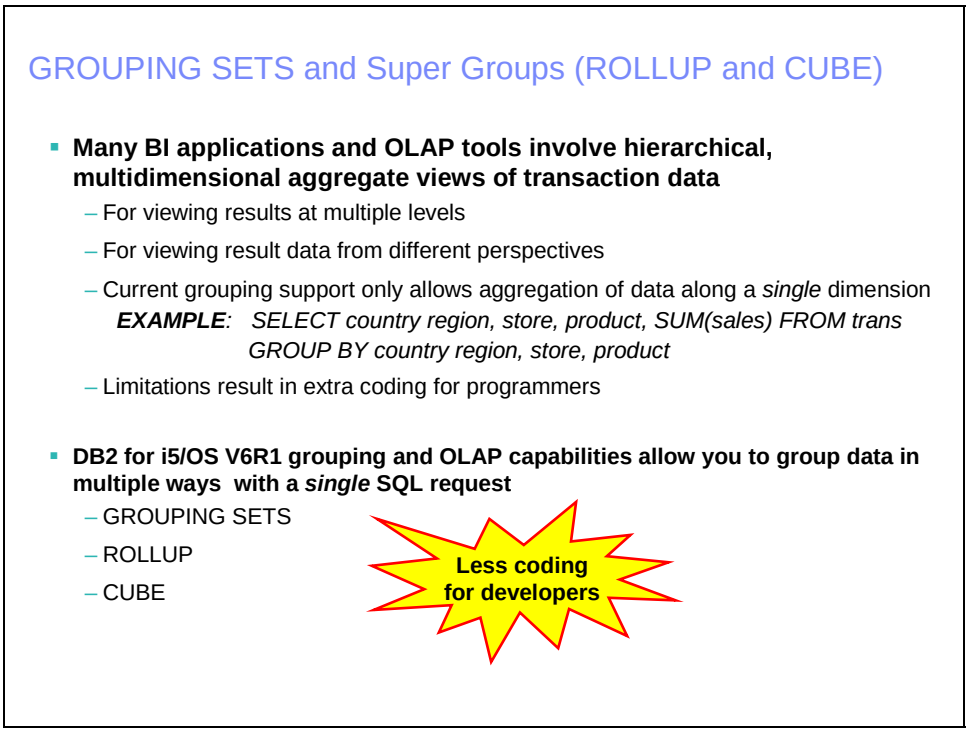
Data-access advancements

The biggest advancements in SQL-based data access and analysis in DB2 for i5/OS V6R1 are the online analytical processing (OLAP) extensions to SQL. (**Note:** The OLAP extensions are covered later on in this section of the course in more detail.)

The new support for INSERT and VALUES statements on the FROM clause make it much easier to port applications that support other DB2 Family members. Support for FULL OUTER JOIN also improves compatibility with the other DB2 products.

SQL developers on the i5/OS platform always benefit from the new additions to the toolbox of built-in SQL functions, and i5/OS V6R1 is no exception. With the intense focus on data privacy within all companies, data encryption continues to be an interest area, so DB2 now has support for the popular Advanced Encryption Standard (AES) algorithm through the ENCRYPT_AES function. Those i5/OS applications that deal with date and time values also benefit from the enhancements that have been added to the VARCHAR_FORMAT and TIMESTAMP_FORMAT functions, as well as the following new functions: MONTHS_BETWEEN, ROUND_TIMESTAMP and TRUNC_TIMESTAMP.

Some of the DB2 for i5/OS V6R1 limits were also increased to give SQL developers more flexibility when constructing SQL requests.



GROUPING SETS and Super Groups (ROLLUP and CUBE)

As business solutions rely more and more on robust data-centric processing, the capabilities of the database management system (DBMS) must continue to step up to the challenge — not just for fast and efficient data access, but also for handling complex query requests that involve sophisticated data manipulations. One of the DB2 for i5/OS V6R1 enhancements is the addition of advanced support for GROUPING SETS and Super Groups; this support allows for OLAP through SQL.

Analysts view data from different perspectives naturally. GROUPING SETS and Super Groups (through CUBE and ROLLUP) allow a user to group and aggregate data in multiple ways — within a single query. Supporting multiple levels of data analysis on a single SQL statement results in less coding for your application developers. Prior to this new support, multiple SQL statements had to be coded and run separately so that data can be presented in hierarchical levels.

ROLLUP

- **ROLLUP on GROUP BY clause results in DB2 returning aggregates for each level of the hierarchy that is implicitly represented in the grouping columns**
  - ROLLUP(Country, Region) will result in the data being summarized at the following levels
    - (Country, Region)
    - (Country)
    - ( ) << represents Grand Total

- Example query**:**
  SELECT country, region, SUM(sales)
  FROM trans
  GROUP BY ROLLUP (country, region)

ROLLUP

The ROLLUP function that is now a part of the GROUP BY clause represents *super grouping,* which is a method for calculating and returning aggregates for each level of the hierarchy that is implicitly represented in the grouping columns. As this chart shows, GROUP BY (Country, Region) returns aggregates at three levels in the final grouping set.

The result set that this example query produces is contained on the next chart to help you better understand the power of the ROLLUP function.

ROLLUP output example

SELECT country, region, SUM(sales) FROM trans
GROUP BY ROLLUP (country, region)

GROUP BY country,NULL

Country	Region	Sum(sales)
Canada	-	100,000
Canada	NW	100,000
U.S.A.	-	3,250,000
U.S.A.	NE	450,000
U.S.A.	NW	940,000
U.S.A.	SE	550,000
U.S.A.	SW	1,310,000
-	-	3,350,000

GROUP BY NULL, NULL

ROLLUP output example

The result set rows that are not circled on this chart are the summary rows that are produced by a simple *GROUP BY country, region* clause. Including the ROLLUP clause results in the inclusion of two additional levels of aggregation into the final result set. Sales are summarized at the country level (top two circled rows); the grand total of sales across all regions and countries is shown in the third circled row).

CUBE

- **CUBE on GROUP BY clause results in DB2 returning aggregates for all possible *distinct combinations* that are represented by the grouping columns**
  - CUBE(Country, Region) results in the data being summarized at the following levels
    - (Country, Region)
    - (Country)
    - (Region)
    - ( ) << represents Grand Total
  - Returns results at multiple intersection points

- Example query**:**
  SELECT country, region, SUM(sales)
  FROM trans
  GROUP BY CUBE(country, region)

Cube

The CUBE function is also considered to be *Super Grouping*. A GROUP BY clause that contains CUBE functions causes DB2 to calculate and return aggregates for all possible distinct combinations that are represented by the grouping columns. In other words, in addition to the grouping along the hierarchy produced by ROLLUP, CUBE also calculates cross-tabular results.

Cubing operations are particularly useful in producing multidimensional results in support of OLAP. Think of a single SQL query that provides information at the various intersections of subjects such as sales figures by department, by store, by location and by date — in effect, producing a *cube* of information.

If you compare the CUBE function on the same two dimensions (Country and Region) as the simple ROLLUP example, you will notice that CUBE returns one additional summary level, Region. Again, an example of the results that are produced by the ROLLUP function are included on the next chart to better help you understand this new feature.

CUBE output example

SELECT country,region, SUM(sales) FROM trans
GROUP BY CUBE (country, region)

GROUP BY NULL, region

GROUP BY NULL, NULL

GROUP BY country, NULL

Country	Region	Sum(sales)
-	NE	450000
-	NW	1040000
-	SE	550000
-	SW	1310000
-	-	3350000
Canada	-	100000
U.S.A.	-	3250000
Canada	NW	100000
U.S.A.	NE	450000
U.S.A.	NW	940000
U.S.A.	SE	550000
U.S.A.	SW	1310000

CUBE output example

Here again, the result set rows that are ***not*** circled are the summary rows that are produced by a simple *GROUP BY country, region* clause. Including the CUBE clause on the grouping requests causes three additional levels of aggregation to be included in the final result set. Sales that are summarized at the country level (shown in the bottom set of circled rows), the region level (shown in the top set of circled rows), and the grand total of sales across all regions and countries (shown in the middle circled row).

GROUPING SETS

- **GROUPING SET on GROUP BY clause enables DB2 to return aggregates for multiple sets of grouping columns**
  - GROUPING SETS((Country, Region), (Country, Store)) will result in the data being summarized at the following levels
    - (Country, Region)
    - (Country, Store)
  - CUBE and ROLLUP can be used in combination with GROUPING SETS
    - **CAUTION:** These types of combinations can result in an exponential growth in the number of grouping sets that are returned by a query, so combine carefully.
- Example query**:**

```
SELECT country, region, SUM(sales)
 FROM trans
 GROUP BY  GROUPING SETS((country, region), (country, store))
```

GROUPING SETS

The GROUPING SET function allows you to specify multiple grouping clauses in a single statement, where DB2 calculates and returns aggregates for each set. Prior to this new grouping support, it was possible to specify only one set of grouping columns. Now, you can provide more than one set of grouping criteria.

As the chart cautions, be very careful when including the CUBE and ROLLUP functions in combination with the GROUPING SET clause. You will probably get many more results than you expect.

In this example, the GROUPING SETS clause is used to two identify two distinct sets of grouping columns: *Country, Region* and *Country, Store*. Therefore, the example query summarizes sales for each distinct grouping of country and region and also summarizes sales for each country and store. The results of this example query are included in the next chart to help you better understand the effect of the GROUPING SETS clause.

GROUPING SETS output example

SELECT country, region, store, SUM(sales) FROM trans
GROUP BY GROUPING SETS ((country, region), (country, store))

GROUP BY COUNTRY, REGION

GROUP BY COUNTRY, STORE

Country	Region	Store	Sum(sales)
Canada	NW	-	100,000
U.S.A.	NE	-	450,000
U.S.A.	NW	-	940,000
U.S.A.	SE	-	550,000
U.S.A.	SW	-	1,310,000
Canada	-	Dougs	100,000
U.S.A.	-	Mariahs	350,000
U.S.A.	-	KMs	770,000
U.S.A.	-	Jennas	400,000
U.S.A.	-	Adrians	500,000
U.S.A.	-	Joshs	300,000
U.S.A.	-	TZs	200,000
U.S.A.	-	Maddies	210,000
U.S.A.	-	Dylans	520,000

GROUPING SETS output example

The results above the dotted line on this chart represent the aggregated results that are produced by the first set of grouping columns (Country and Region). The results below the dotted line represent the summarized results from the second set of grouping columns (Country, Store).

GROUPING SETS and Super Groups considerations

- **Use the GROUPING function to determine if null values are from underlying data or DB2 group processing**
  - Function returns **1** if the grouping column contains a NULL value that was produced by GROUPIMG SET or Super Group processing
  - Function returns **0** if grouping column contains *real* GROUP BY value

  EXAMPLE: *SELECT country,region, store, GROUPING(store), SUM(sales)*
  FROM trans WHERE transYear=2006
  GROUP BY GROUPING SET((country, region),(country, store))

- **Performance considerations**
  - The SQE query optimizer contains patented technology that allows DB2 to internally compute multiple aggregates in a *single pass* of data
  - GROUPING SETS and Super Sets help the optimizer by creating indexes that cover all of the grouping columns (in addition to any local, equal selection predicates)
    - Best index keys for sample query shown here: (transYear, country, region, store)
    - Index Advisor has been enhanced to support new grouping capabilities ,too.

GROUPING SETS and Super Groups considerations

You might have noticed in the Super Group and Grouping Set examples, that DB2 manually assigns *null* to some of the column values, depending on the level of aggregation that is computed. What happens when the underlying query data also contains null values? Can you tell the difference between *real* null values and null values that DB2 assigns? The answer is, “Yes, the Grouping scalar function provides this capability. The Grouping scalar function returns a value of **1** if the null column value is a result of the DB2 GROUPING SET or Super Group processing, and returns a **0** if the underlying data in a group column contains an application-supplied null value.

Computing multiple summaries usually requires a database engine to read the underlying data several times. Reading the same set of data multiple times for a query is usually not ideal from a performance perspective. Luckily, that is not the case with DB2 for i5/OS because the SQL Query Optimizer (SQE) contains patented technology that, the majority of the time, allows DB2 to compute multiple aggregates in a single pass of data. Even with these patented optimization techniques, the optimizer still needs indexes to be created to help it provide your application with the best possible performance. Therefore, it is best to follow the indexing recommendations and performance considerations listed on this chart.

Data-definition enhancements

- **CREATE TABLE enhancements**
  - New data types
    - DECFLOAT
    - NCHAR, NVARCHAR, NCLOB (UTF-16)
  - New column attributes
    - Hidden column
    - Row-change timestamp
- **Unsupported-syntax tolerance**
- ***Automatic* encryption with ASP-level encryption (AES)**
- **Derived SQL indexes**
- **User-defined function (UDF) improvements**
  - ALTER FUNCTION for simpler maintenance
  - Common Table Expressions on RETURN clause
- **Miscellaneous**
  - Improved metadata with COMMENT and LABEL enhancements
  - Statement-level Instead Of triggers
  - NEXT_IDENTITY_VALUE for table available in QSYS2.SYSPARTITIONSTAT

Data-definition enhancements

SQL table definitions are also more portable to DB2 for i5/OS with the addition of several new data types and column attributes. The National Character data types (NCHAR, NVARCHAR, and NCLOB) provide a standard way of defining a Unicode column with UTF-16 encoding. The decimal float data type, DECFLOAT, is a new numeric data type that has been added for compatibility with the other DB2 products; it combines the attributes of decimal and float with extended accuracy. The new column attributes are covered in more detail later in this course.

The portability of SQL database-creation scripts to DB2 for i5/OS is also simpler in DB2 for i5/OS V6R1 with new tolerance of unsupported syntax. The details of this enhanced syntax tolerance is covered later in this course.

Encryption is a hot topic these days as companies try to protect consumer and business data. On prior releases, data encryption required application changes to protect the data. DB2 for i5/OS V6R1 supports the creation of an encrypted auxiliary storage pool (ASP) to automatically encrypt data as it is written to disk in the ASP (and decrypted when the data is read from disk). There is encryption support for both user ASPs and independent auxiliary storage pools (IASPs); however, this support does not allow for the encryption of an existing ASP. Thus, you must move any objects in an existing ASP that require encryption to a newly created ASP on i5/OS V6R1. Some minor modifications to your environment and the application might be necessary to access data that is stored in an IASP.

The ability to create derived SQL indexes is one of the most interesting data-definition enhancements in DB2 for i5/OS V6R1. This new capability is covered in-depth in the *Performance enhancements* section of this course because it provides great flexibility when tuning complex SQL requests.

Other miscellaneous enhancements include the improvements listed on this chart for user-defined functions (UDFs) and the upgraded LABEL and COMMENT statements that now allow you to provide descriptive metadata for constraints, routines and triggers.

New column attributes: Hidden timestamp

- **Implicitly Hidden Timestamp and Row Change Timestamp attributes enable DB2 to track row-level changes for you**
  - Clauses can be used independently
  - Attributes are frequently combined to create a Hidden Timestamp column
  - Hidden Timestamp columns are used to implement an Optimistic Locking scheme
    - Read values in a row without an update lock
    - Perform calculations on fetched values
    - UPDATE same row with new values, check Timestamp column to see if row has changed

```
CREATE TABLE tickets(
 ticket_ord     INTEGER,
 ticket_qty     INTEGER,
 ticket_event  VARCHAR(10),
 ticket_ts    TIMESTAMP NOT NULL
              IMPLICITLY HIDDEN
              FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP);
INSERT INTO tickets VALUES(1,11,'mvGAME1'),
                           (2,8,'ihGAME4');
```

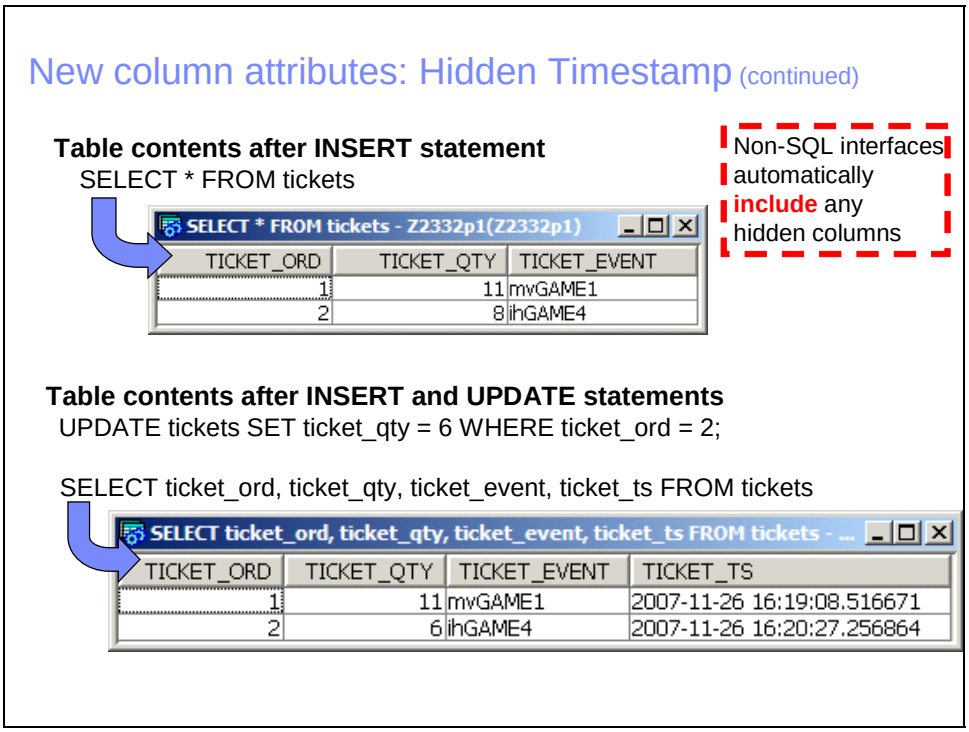
NOTE: Only three column values passed on INSERT

New column attributes: Hidden timestamp

Both Microsoft SQL Server and MySQL tables include support for what is known as a *Hidden Timestamp* or A*utomatic Timestamp* column that was difficult to emulate when porting to DB2 for i5/OS prior to i5/OS V6R1. When a table in one of these database products includes a timestamp column, the database engine automatically updates that timestamp column each time that a row is inserted or updated. This automatic update behavior provides applications with an easy mechanism to determine if a table row that was read earlier has changed — before attempting to update that same row in the table. Determining if the row has changed is done simply by comparing the value of the Timestamp column when the row was read with the current value of the Timestamp column in that same row of data. This timestamp comparison is central to applications that use an optimistic-locking approach and is easy to implement with the Hidden Timestamp column support in DB2 for i5/OS V6R1.

The example at the bottom of this chart shows how to create a table with a Hidden Timestamp column. The Create Table statement shows the new syntax for hiding a column with the IMPLICITLY HIDDEN clause on the ticket_ts column definition. This Hidden clause indicates that the column is not visible on any SQL statements, unless it is explicitly referenced by name.

Notice that the INSERT statement that references the tickets table with the Hidden Timestamp includes only three column values. The value for the Hidden Timestamp column does not need to be applied because DB2 is responsible for setting that column on INSERTs and UPDATEs.



New column attributes: Hidden Timestamp (continued)

The behavior of the Hidden Timestamp column, ticket_tx, is shown here with the output for the first SELECT statement on this page. The ticket_ts column is not part of the result set because it is not explicitly referenced on the SELECT statement.

The FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP clause on the ticket_ts column signifies that DB2 changes the ticket_ts column value each time that a row is inserted or updated into the table. After running the UPDATE statement shown on this chart, the effect of the Timestamp change clause is shown with the SELECT statement.

Notice that DB2 automatically assigns and changes the value of the ticket_ts column, even though the INSERT statement (on the prior chart) and the UPDATE statement contain no references to the ticket_ts column.

Columns that are defined with the Hidden clause are automatically returned to any applications that use non-SQL interfaces. Thus, the Hidden clause is only enforced when SQL-based data access is performed. The Row Change Timestamp clause does apply to non-SQL interfaces; thus, a timestamp column is updated when a non-SQL interface changes existing rows or inserts new rows.

Unsupported-syntax tolerance

- **SQL parser is enhanced to tolerate unsupported syntax that is not needed on DB2 for i5/OS**
  - Unique architecture of i5/OS enables some SQL statements and clauses to be ignored
  - Speeds up porting process

Examples:
CREATE TABLESPACE TS1 MANAGED BY DATABASE USING (device '/dev/rcont $N' 20000)
SQLSTATE: 01505 / SQLCODE: +143
Message: Statement CREATE TABLESPACE ignored.

CREATE TABLE newtab1 (c1 **INT)** INDEX IN ts1
SQLSTATE: 01680 / SQLCODE: +20367
Message: Clause INDEX IN ignored.

CREATE TABLE t1 (c1 **INT) IN** ts1
SQLSTATE: 42704 / SQLCODE: -204
Message: TS1 in L1 type *NODGRP not found.

Unsupported-syntax tolerance

Often, the SQL scripts for other DB2 databases contain low-level configuration syntax (for example, table-space options) on the SQL data-definition language statements that prevent the scripts from running on DB2 for i5/OS until you remove the unsupported syntax from the database scripts. As a result, DB2 for i5/OS V6R1 ignores the syntax and options that DB2 for i5/OS will never support. This is because the DB2 for i5/OS database engine and operating system automatically handle many low-level database administration tasks that require manual configuration with the other DB2 products.

This chart contains examples of statements with the ignored syntax (circled on this chart). An SQL warning is returned to flag ignored syntax on DB2 for i5/OS. Porting is faster without the need to delete unnecessary syntax from database-script files.

Availability and recovery enhancements

Availability and recovery enhancements

Next, you will learn some details about the i5/OS V6R1 enhancements that make it easier to enhance your availability and recovery plans.

Database availability and recovery

- **DB2 engine improvements**
  - Improved scalability of *online* reorganize
  - Transaction quiesce for cross-site mirroring (XSM) switchover
  - Deferred dependent-object processing for restoring databases
    - Enables restore to complete when Index, LF, or MQT and the underlying table reside in different libraries
    - Can minimize index rebuilds
    - New RSTDFROBJ command
  - Object identifier preservation
    - File identifier on CRTDUPOBJ and CPYLIB
    - Member identifier and change date on CPYSRCF and CPYF
  - SQL package integrity

Database availability and recovery

The availability and recoverability of DB2 for i5/OS has long been characteristic of the System i database — differentiating it strongly from the competition. IBM continues to distinguish these characteristics with additional investments in DB2 for i5/OS V6R1.

The ability to do online (concurrent) reorganization was introduced in i5/OS V5R4. However, the i5/OS V5R4 support required an exclusive lock to be obtained during the reorganization to give back any reclaimed storage to the operating system. This lock reduced the time that a DB2 object could be *online* during the operation. DB2 for i5/OS V6R1 eliminates this issue by using a less-obtrusive serialization mechanism.

Cross-site mirroring (XSM) is one technology that System i administrators use to copy production data to a backup server. DB2 for i5/OS V6R1 adds database-transaction quiescing to prevent partial transactions from being mirrored to a backup server. The quiese-point processing is very similar to the i5/OS algorithm used in the Save While Active support.

Restoring database objects is a key part of the recovery process after system failure. Database-restore processing at times is slowed by users who do not follow IBM guidelines of creating dependent objects such as an index, logical file (LF) or materialized query table (MQT) in the same schema (or library) as the underlying media. When this recommendation is not followed, database-restore operations can fail — depending on the order in which libraries are restored onto the system. If you restore an index prior to restoring the underlying table, the restoration of the index fails, and you must manually retry the index restoration after restoring the table. This manual retry also involves reloading the save media.

DB2 for i5/OS V6R1 adds deferred dependent-object processing to enable the system to be responsible for retrying the restoration of dependent objects — without requiring the need to reload. This new DB2 for i5/OS V6R1 support does require that you specify the new DEFERID on the restore operation and that you use the new Restore Deferred Object command (RSTDFROBJ) to start the dependent-object processing. The IBM Backup Recovery and Media Services (BRMS) support has also been updated to take advantage of this new dependent-object support. Despite these DB2 for i5/OS V6R1 enhancements, IBM still recommends that you create dependent objects (such as Index and LF) in the same library as the table.

Preservation of File Identifiers on the Copy Library(CPYLIB) and (Create Duplicate Object (CRTDUPOBJ) commands is helpful for application developers and change-management tool vendors who want to avoid the program recompiles that were required in the past when these commands generated new identifiers.

The DB2 engine is also enhanced to better preserve the integrity of SQL Package objects. These enhancements minimize the need for administrators to delete and recreate packages.

Database availability and recovery

- **Journaling enhancements**
  - Library-level journaling with new STRJRNLIB command
    - Enables *automatic* journaling of DB2 objects
    - Similar function to the QDFTJRN data area
  - *ALL support for Start Journal commands
  - DDL journaling improvements
    - Merging of APYJRNCHG and APYJRNCHGX
    - Creation of SQE column statistics
    - Improved Change Member support

Database availability and recovery

Journaling (or logging) is a key component of any business solution that needs to be highly available. The i5/OS journaling capabilities have been enhanced greatly over the past couple of releases, and DB2 for i5/OS V6R1 is no exception. The DB2 for i5/OS V6R1 journal enhancements follow a key theme of making it easier for users to journal their objects.

Instead of requiring you to start journaling for individual objects, the new Start Journal Library (STRJRNLIB) command allows for the automatic journaling of objects that are moved, created or restored into a journaled library. This automatic-journal capability is similar to the capabilities delivered with the QDFTJRN data area that was introduced in i5/OS V5R4.

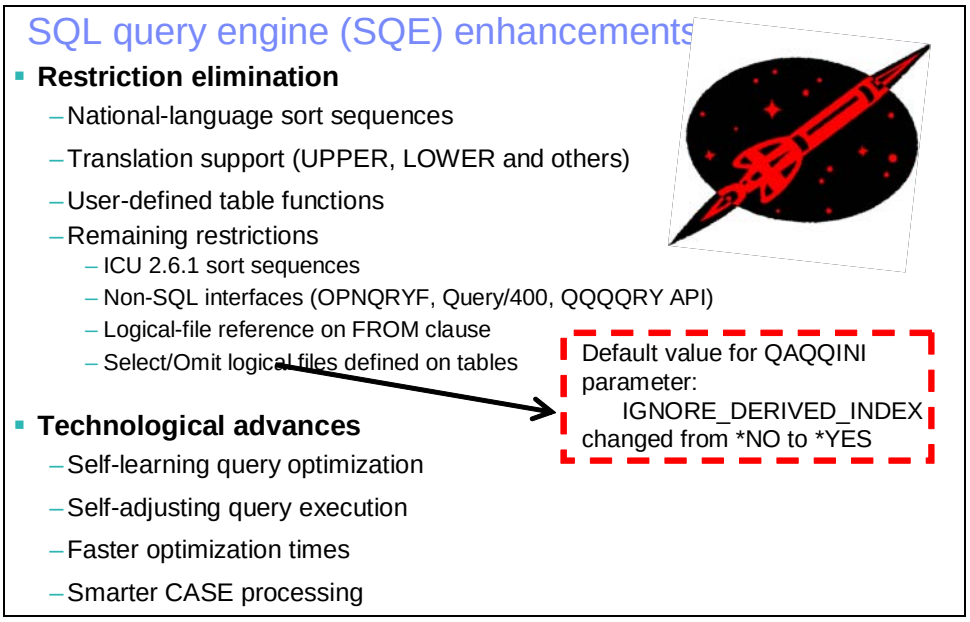
In a similar fashion, the Start Journal Physical File (STRJRNPF) and Start Journal Object (Start Journal Object) commands are enhanced to support an *ALL value for the Object parameter. Being able to start journaling for all objects in a library makes it much easier to restart journaling after a restore operation that restores the objects into a different library from the one in which they were saved.

Improved journaling of data-definition language (DDL) changes makes it easier to recover databases that have had new objects created or existing objects altered since the last database backup. These enhancements also improve of the robustness of high-availability solutions that rely on journal entries to keep the backup system up-to-date. For example, being able to automatically collect a column statistic on a backup system eliminates the need for administrators to manually collect column statistics on the backup system. The better the synchronization is between the backup system and the production system, the better performance is on the backup system when a system failure forces usage of the backup system.

Performance enhancements

Performance enhancements

Next, you will learn some details about the i5/OS V6R1 enhancements that are important to having high-performance DB2 for i5/OS applications.



SQL Query Engine (SQE) enhancements

The SQL Query Engine (SQE), first introduced in IBM OS/400® V5R2, has been one of the key IBM initiatives for taking SQL performance on DB2 for i5/OS to new heights. This performance advancement for SQL workloads continues in DB2 for i5/OS V6R1 with the elimination of two major roadblocks that prevented the usage of the SQE: National Language Sort Sequences (NLSSs) and functions that rely on low-level translation function. Commonly used, built-in SQL functions (such as Upper and Lower) use this low-level translation capability. The NLSS support in i5/OS is frequently used in overseas markets where the application requires the character data to be sorted in a manner that matches the local language, instead of the default *HEX ordering, which closely mirrors the sorting of the English alphabet. With these key functions now supported by SQE in DB2 for i5/OS V6R1, the most common items that prevent the usage of SQE in DB2 for i5/OS V6R1 are listed on this chart.

Although the SQE optimizer still cannot use Select or Omit logical files when building query plans, IBM did change the default value for the Ignore_Derived_Index QAQQINI parameter to enable more SQE usage. This QAQQINI parameter was first added back in i5/OS V5R3 to allow the SQE to be used in environments where SQL statements refer to DB2 objects that are created with data-definition specifications (DDS). Prior to DB2 for i5/OS V6R1, the default value for this parameter was *NO, which caused the SQE optimizer to reroute the running of any SQL request to the Classic Query Engine (CQE) when a derived logical file was encountered during the query-optimization process. The default value in DB2 for i5/OS V6R1 for the Ignore_Derived_Index QAQQINI parameter has been changed to *YES. This value allows the SQE optimizer to ignore any keyed logical files that contain Select or Omit criteria during optimization, instead of rerouting the execution to the CQE.

Even though eliminating barriers to SQE usage is important, the most interesting DB2 for i5/OS V6R1 enhancements are those new capabilities that leverage the extensible architecture of the SQE. A prime example is the first DB2 for i5/OS foray into self-learning query optimization. A self-learning query optimizer can analyze poorly performing query plans and dynamically adjust its internal algorithms, based on feedback, to select a better query plan on future runs. In DB2 for i5/OS V6R1, the SQE optimizer automatically analyzes poorly performing query plans to

determine its I/O characteristics and the record-retrieval patterns. It then compares them to the values that are used during optimization of the query. If the query optimizer detects significant mismatches, then the optimizer modifies its assumptions and algorithms during the next execution of that SQL request to generate a better plan. To complement the learning optimizer, the DB2 runtime engine is also equipped with adaptive technologies that allow the plan of a currently running query to be modified on the fly — to improve the efficiency of the query for the remainder of the run.

SQL derived indexes

- **SQL key definitions support expressions, functions and operators to enable more usage of indexes by the query optimizer on complex queries**
  - Fully supported by SQE optimizer (limited support by CQE)
  - *EXAMPLES*:
    CREATE INDEX ix_TotalSalary ON employees (Sales + Bonus)
    CREATE INDEX ix_FullName ON employees (CONCAT(CONCAT(FName,' '), LName))
- **Great for improving performance of case-insensitive searches**
  SELECT cust_id, cust_phone FROM customers
  WHERE UPPER(company_name) = 'ACME'
  CREATE INDEX ix_uCompName ON customers(UPPER(company_name))

  Fast and Simple
- **Create Index statement also supports the ability to create sparse indexes (that is, Select and Omit criteria) but there currently is *no* optimizer awareness**
  - Short-term value: provide SQL replacement of DDS Select/Omit logicals
  - *EXAMPLE*: CREATE INDEX cust_ix1 ON customers(cust_id) WHERE activCust='Y'

SQL derived indexes

The other key advancement in DB2 for i5/OS indexing technology is a feature called *SQL derived indexes* (or functional indexes). In some ways, you can view this as the i5/OS SQL interface catching up with DDS by allowing SQL indexes to be created with expressions and selection criteria — just as i5/OS and OS/400 developers have been doing for years on keyed logical-file definitions. Although that is partly true, SQL derived indexes take these capabilities even further with the ability to specify SQL built-in functions as part of the key expression, as you can see in the first two examples on this page.

The Upper function will probably be one of the more popular functions used in SQL-derived indexes. Application developers often use the Upper function to implement case-insensitive searches, such as the one shown here by the SELECT statement. By forcing all of the company names to be converted to upper case, the query returns all occurrences of ACME, even if the data-entry clerks entered that company name differently (for example, *Acme* or *acme*).

Although this type of case-insensitive search had no functional issues on DB2 for i5/OS, performance problems can arise because the UPPER function prevented the query optimizer from using an index to speed up this search. The new SQL derived-index support allows the index to be created with the Upper function in the key definition. It also allows the SQE optimizer to use this index to quickly retrieve customer data, because the key expression on the index exactly matches the column expression in the case-insensitive query.

Some application developers have successfully used shared-weight sort sequences to solve the performance issues that are associated with case-insensitive searches. However, that approach requires more complex configuration when creating your database and applications. In addition, sort sequences cannot be used to provide the optimizer with an index for more complex expressions such as:

```
WHERE UPPER(CONCAT(FirstName,LastName)) = 'ALBERTYOUNG'
```

A large majority of the SQL built-in function can be included in the key definition of an SQL derived index, consult the SQL Reference for the complete list.

The last derived-index example on this chart shows the key definition that also contains selection criteria. This is known as a *sparse index* because the index does not contain a key value for every row in the table. Even though you can create this sparse SQL index on DB2 for i5/OS V6R1, it must be noted that the SQE optimizer currently does not have the ability to use a sparse index in a query implementation — it can only choose SQL derived indexes with key expressions. IBM plans to enhance the SQE optimizer in the future with the ability to use sparse indexes. However, in the short-term, you might consider using the key selection criteria to recreate existing Select and Omit logical files as SQL indexes.

Additional performance enhancements

- **SQL performance**
  - Full Open performance (that is, first execution)
  - Stored Procedure CALL caching
  - Compression of variable-length columns in result set
- **Miscellaneous**
  - More efficient Alter Table
  - Faster reorganize for BLOBs and CLOBs
- **Database Monitor**
  - Enhanced support for OPNQRYF and Query/400 requests (1000 record)
  - Enhanced host-variable and parameter-marker value collection (3010 record)
  - New filters for Query Governor and TCP/IP ports
- **New QAQQINI options**
  - SQL_FLAGGER: Identifying nonstandard SQL syntax
  - SQL_STMT_REUSE: Customize SQL package behavior
  - SQL_PSEUDO_CLOSE: Override default ODP reuse algorithm
  - QSYS2.OVERRIDE_QAQQINI procedure for customizing QAQQINI usage

Additional performance enhancements

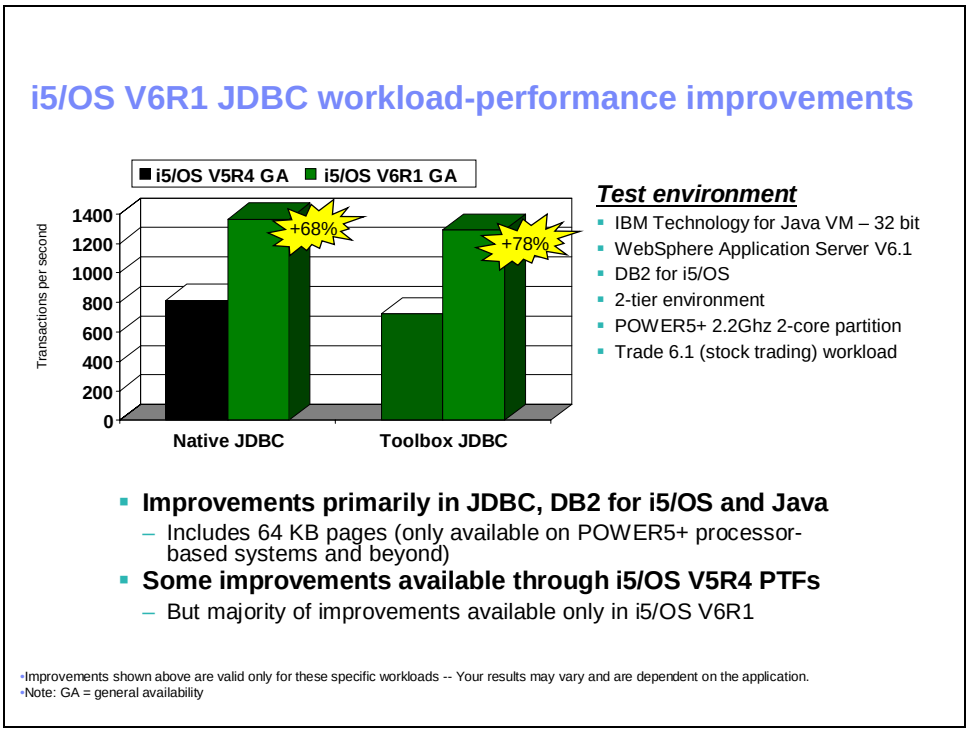
SQL workloads on i5/OS also run faster on i5/OS V6R1 with the streamlining of the SQL Full Open code path. *Full Open* refers to the processing that DB2 must perform the first and second time an SQL statement runs within a connection (or job). Some tests show a 10 percent performance improvement. Applications that use stored-procedure calls also receive a nice boost from improved caching of repeated stored-procedure calls, which can be quite common in some applications. ALTER TABLE and Reorganize commands also run faster in i5/OS V6R1 as a result of the miscellaneous performance enhancements listed on this chart.

IBM has enhanced the data that is collected by the Database Monitor to complement the enhancements to the graphical-analysis interfaces. One of the biggest additions is the monitor's ability to collect detailed monitor data for Query/400 and OPNQRYF requests. With this addition, the System i Navigator Visual Explain tool and SQL Performance Monitor analysis reports can be used for the first time to analyze and tune the performance of Query/400 and OPNQRYF requests. Enhancements to the collection of host-variable and parameter-marker values also makes it easier to reconstruct the exact SQL statement request from monitor data (in particular, for INSERT, UPDATE and CALL statements).

New filter capabilities on TCP/IP port values and Query Governor limits also give administrators better granularity in controlling the amount of data that the monitor collects.

There are three new QAQQINI parameters in DB2 for i5/OS V6R1. The SQL_FLAGGER parameter is useful to programmers who need to determine if they are using non-standard SQL syntax. You can use the SQL_STMT_REUSE and SQL_PSEUDO_CLOSE parameters to modify the performance behavior of the SQL engine. However, it is best to use the default values for these parameters for the majority of the time. A deep knowledge of the SQE is recommended before changing these parameters from their default values.

The Override_QAQQINI stored procedure is available on DB2 for i5/OS V6R1 to make it easier to apply a partial list of QAQQINI parameter settings to a single job on the system. You can use this stored procedure to create and populate a temporary QAQQINI file.



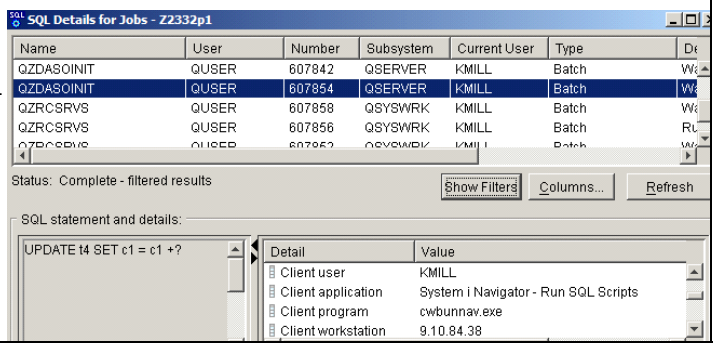
I5/OS V6R1 JDBC workload-performance improvements

This chart shows the impact that some of these DB2 performance enhancements had on JDBC workloads that run on i5/OS V6R1 — along with additional enhancements that have been made to the JDBC drivers.

Simplifying performance analysis

- **Client special registers are added to link SQL request with the application**
  - CURRENT CLIENT_ACCTNG
  - CURRENT CLIENT_APPLNAME
  - CURRENT CLIENT_PROGRAMID
  - CURRENT CLIENT_USERID
  - CURRENT CLIENT_WRKSTNNAME
  - APIs available for application instrumentation, some IBM middleware is instrumented
    - SQLESETI i5/OS API (Stored procedure example at: ibm.com/systemi/db2/db2code.html)
    - CLI SQLSetConnectAttr() function
    - JDBC setClientInfo connection method
- **Values are accessible with Database Monitor and *SQL Details for Job* tool**
  - 1000 record type
  - Register values can also be retrieved with SQL

```
SELECT
CURRENT
CLIENT_APPLNAME
  FROM anyTable
```



Simplifying performance analysis

Tuning the performance of DB2 for i5/OS can be quite challenging when SQL requests are submitted from remote clients, such as browsers and .NET applications. A system administrator can easily detect i5/OS server jobs that consume extra resources because of a long-running SQL statement. However, it is quite difficult to trace back and determine which remote client or program submitted the SQL request. Thus, in i5/OS V6R1, IBM introduces the five client special registers listed here to allow applications to use them in an effort to make this problem-determination exercise easier.

Application developers can instrument their applications to set these values with one of the APIs listed on this chart. After the application sets these registers, administrators can retrieve these settings by collecting database-monitor data or by using the System i Navigator *SQL Details for a Job* task — by right-clicking the **Database** object in the navigation tree. You can see the output from the System i Navigator task in the screen capture shown on this chart. In prior releases, this Navigator task was known as *Current SQL for a Job*.

Look at the SQL Details output closely and notice that IBM has instrumented some of its own tools and drivers to assign values to these new client special registers. For example, notice that user KMILL ran the UPDATE statement from the System i Navigator Run SQL Scripts interface.

Ease-of-use and management enhancements

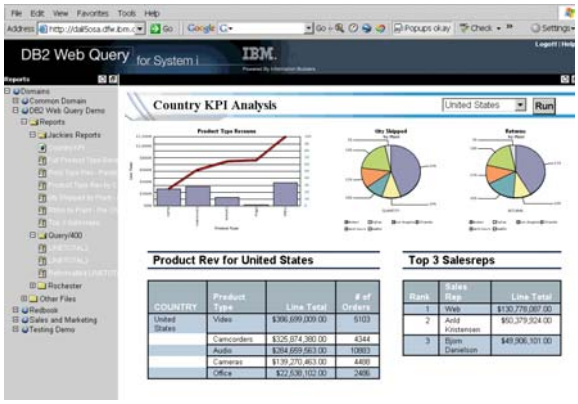
Ease-of-use and management enhancements

Next, you will learn some details about the i5/OS V6R1 enhancements that make it easier for analysts and administrators to manage their systems.

Enhanced DB2 for i5/OS tools

- **IBM System i Navigator**

- **IBM DB2 Web Query for System i – 2008 planned enhancements**
  - Runtime user licensing
  - Report scheduling and distribution
  - Solution developer kit (SDK) for application integration

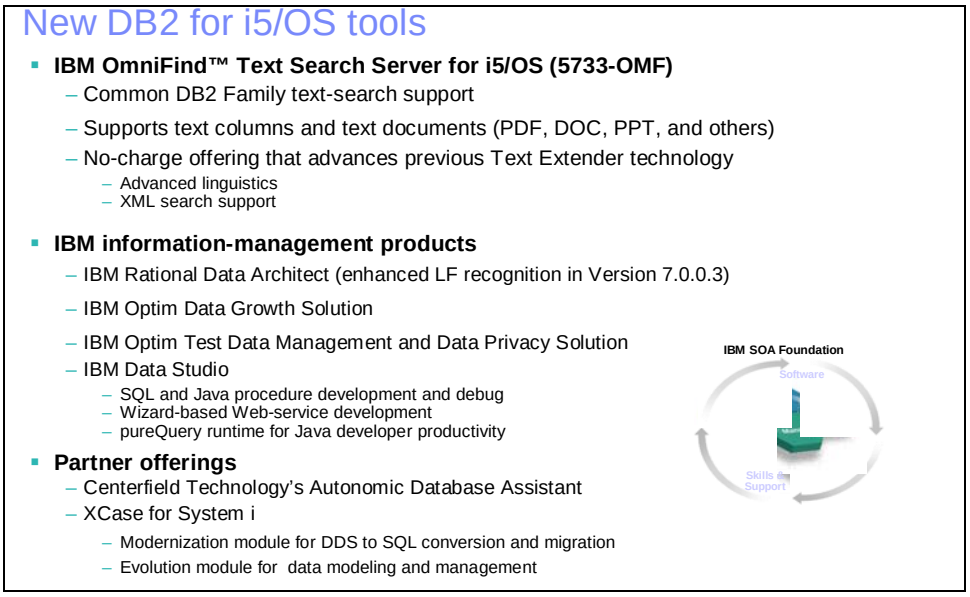


Enhanced DB2 for i5/OS tools

IBM and its Business Partners continue to enhance existing tool sets and build new tools to make it even easier to manage and use DB2 databases.

IBM System i Navigator for i5/OS (formerly iSeries Navigator), which is the graphical-management interface for DB2 for i5/OS V6R1, contains a number of enhancements that are covered in more detail later in this course.

System i developers and users also benefit from the multiple enhancements that IBM plans to add to DB2 Web Query in 2008. First, IBM plans to add more flexible pricing terms to accommodate both users and developers when running reports. In addition, IBM is working on new interfaces that are designed to make it easier for developers to leverage and integrate DB2 Web Query reports into their applications. These improvements are planned to include report scheduling and automated report distribution.



New DB2 for i5/OS tools

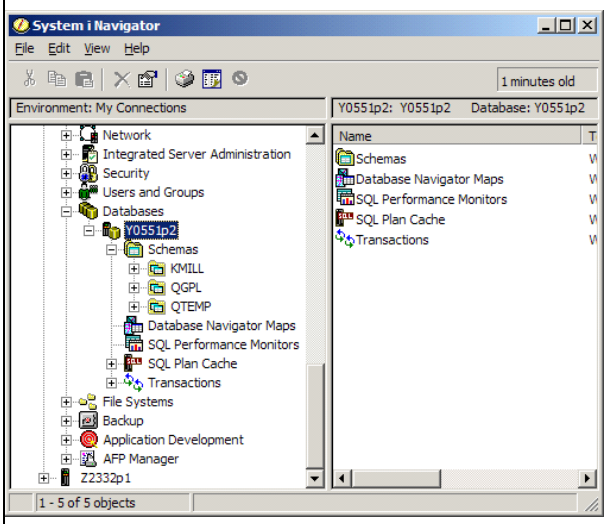
The IBM OmniFind™ Text Search server brings common DB2 Family text-search support to DB2 for i5/OS. This new text-search server supports the searching of text within columns as well as text documents. This no-charge offering advances the DB2 Text Extenders for i5/OS support. You will find the biggest improvements in the support for rich-text documents (for example, Adobe® PDF, Microsoft Word and PowerPoint), search support for XML documents, and advancements in linguistics. These linguistic advancements include better support for matching based on word variations (for example, a search for the word *mouse* also finds the word *mice*).

The IBM software portfolio contains a number of new tools that you can use with DB2 for i5/OS:

- **IBM Rational Data Architect** is a data-modeling tool for use with DB2 for i5/OS databases. It supports DB2 objects that are created from the non-SQL interfaces (for example, logical files).
- **IBM Optim Data Growth Solution** lets you safely archive data that is not relevant to your daily business operations to a secure archive. This solution makes the archive easily accessible if you need the old business data for compliance or audit purposes.
- **IBM Optim Test Data Management and Data Privacy Solution** simplifies the process of creating and maintaining streamlined database-test environments. Additionally, the data privacy helps you protect your customers' privacy by masking sensitive data that is stored in your test databases.
- **IBM Data Studio** is an enhanced tool set that simplifies the creation and debugging of SQL and Java stored procedures, along with Web services. In addition, it provides the pureQuery run time to improve the productivity of Java developers who need to access DB2 for i5/OS data.

At the bottom of this chart, you will see new offerings from third-party tool providers that can automate the performance tuning of DB2 for i5/OS and help move your legacy database definitions from DDS to SQL.

System i Navigator enhancements



OnDemand Performance Center
- Spreadsheet integration
- Column customization
- Index Advice Condenser (PTF)
- Fast monitor summary compare
- Plan Cache Resize and Event Monitor
- Enhanced plan-cache analysis
- Visual Explain while running
- Special client register

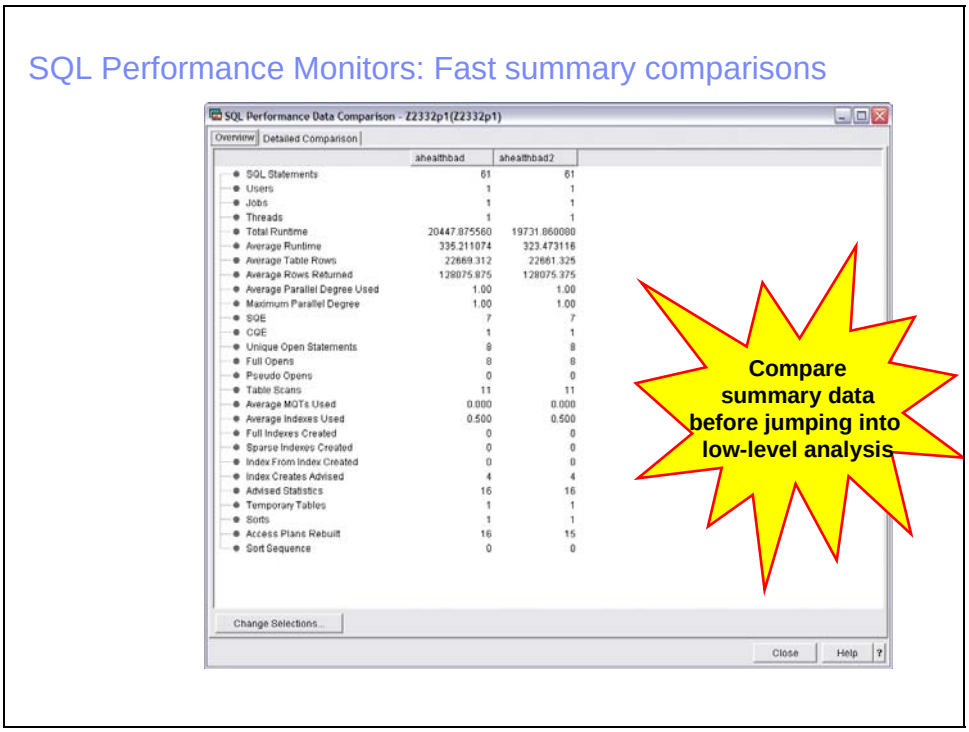
Run SQL scripts
- Improved font sizer
- SQL syntax flagger
- Source-member integration
- UTF-8 support

Database management
- Show object locks (WRKOBJLCK)
- Schema-level index evaluator
- Enhanced SQL details for job

Health Center
- Environmental limits
- Journal and journal receivers
- Procedures and packages
- Table activity

System i Navigator enhancements

Major improvements have been made to the DB2 support in System i Navigator in past releases and i5/OS V6R1 is no exception. This page includes a partial list of enhancements that are delivered with i5/OS V6R1. Similar to i5/OS V5R4, the biggest improvements relate to the collection of graphical-performance tools that are known as the DB2 OnDemand Performance Center. The most significant DB2 enhancements to System i Navigator are covered in the charts that follow.

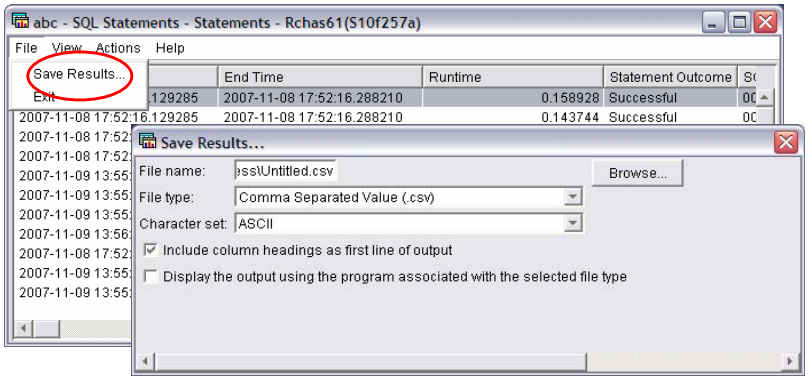


SQL Performance Monitors: Fast summary comparisons

The usability of the SQL Performance Monitors in System i Navigator took a major step forward in i5/OS V5R4 and that continues with i5/OS V6R1. The i5/OS V5R4 support includes a utility to compare database-monitor collections, but the comparison occurs at the SQL-statement level. Many times, analysts first want to compare monitor collections at a high level before deciding if a statement-level comparison is necessary.

As you can see on this chart, i5/OS V6R1 now includes a summary-level monitor comparison that compares higher-level metrics, such as the number of SQL statements and the total run time for the SQL statements.

SQL Performance Monitors: Shareable analysis

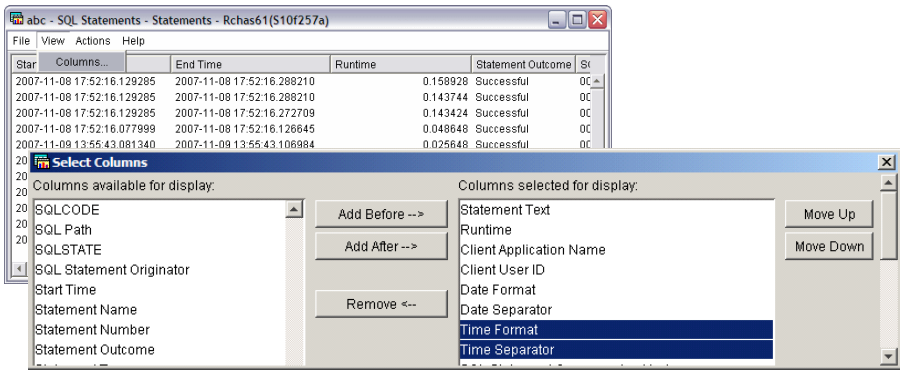


- **Save results into spreadsheets or other file formats (for example, comma-separated value, or CSV)**
- **Sharing results is also available with the Run SQL Scripts interface.**

SQL Performance Monitors: Shareable analysis

The SQL Performance Monitors Analyze task makes it quite easy for a user to identify a set of SQL statements that have performance issues. However, sharing those SQL statements with an administrator or performance analyst was quite difficult prior to i5/OS V6R1 because the options were either to perform a primitive copy-and-paste operation or to tell the administrator or analyst how to run the same report. i5/OS V6R1 simplifies this operation with a new Save Results task that lets you save the results directly — in a wide variety of output formats. This set of screen captures shows this new interface. The dialog not only allows you to save the report data into a shareable format, but also provides an option for launching the PC application (in this case, Microsoft Excel) that is associated with this file type.

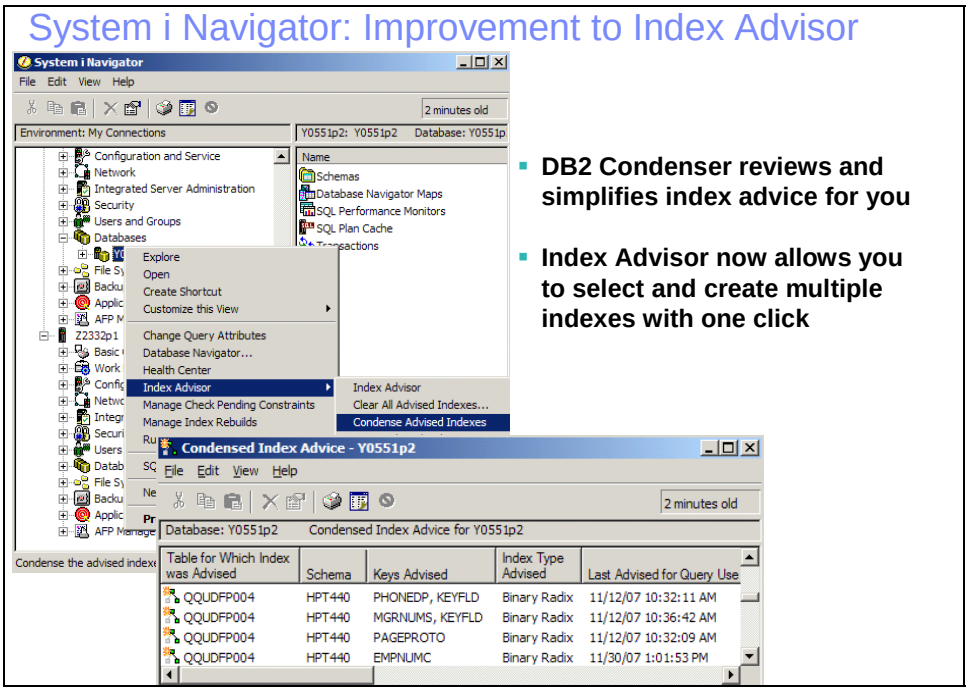
SQL performance monitors: Customizable analysis



- **Simplifies performance analysis by allowing you to concentrate on the most important columns**
- **Customizations are remembered across sessions and servers**

SQL Performance Monitors: Customizable analysis

Column customization is another i5/OS V6R1 enhancement that makes System i Navigator output easier to read and analyze. The SQL Performance Monitor analysis reports are very thorough in returning every piece of information that might affect DB2 performance. The downside of these complete reports is that the numerous columns of returned data can overwhelm the administrator or analyst. Some reports return almost 100 columns of data. In addition, the default ordering of the columns in the reports might slow your ability to find most performance information. For example, some reports require the user to scroll two or three screens to the right to find the SQL-statement text. The column-customization support in DB2 for i5/OS V6R1 solves both of these problems by letting you reorder or remove columns from the output display. You accomplish this customization through the Select Columns pop-up window that is displayed on this chart. Column customization is also available on other interfaces, such as the Plan Cache, Show Indexes and Index Advisor interfaces. After you customize an interface, the reporting tools remember those settings on future reports.



System i Navigator: Improvement to Index Advisor

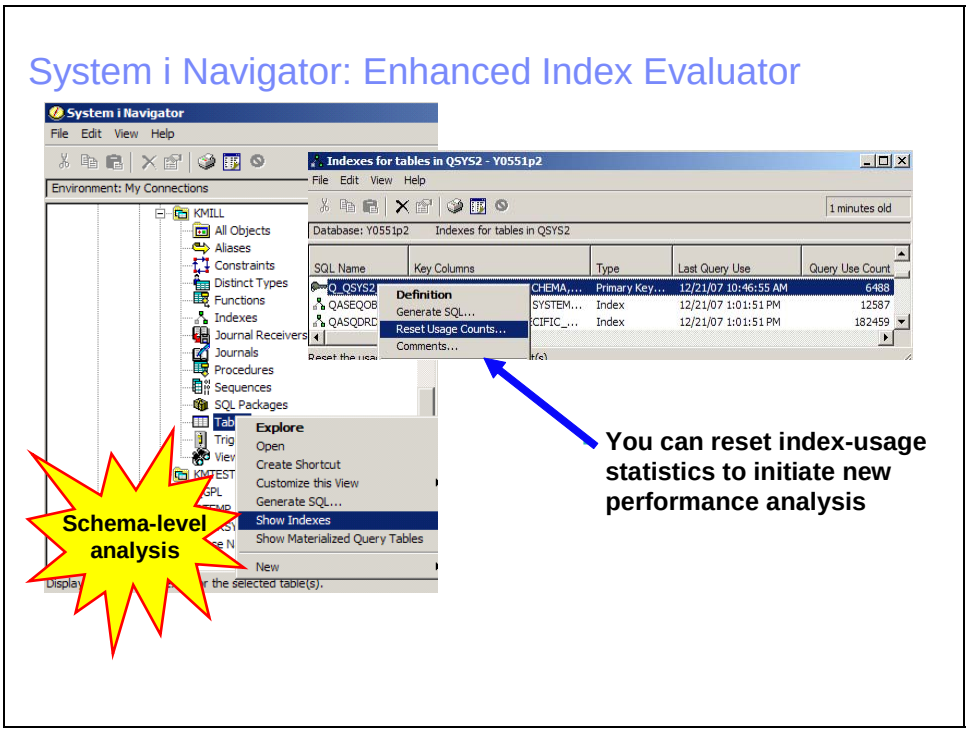
The usability of the index analysis tools in the System i Navigator is also improved with i5/OS V6R1. Users really like the real-time index advice that is provided by the system-wide Index Advisor (which was added in i5/OS V5R4). However, the interface for creating indexes that are based on the advice has not been as well received. If the Index Advisor suggested 10 indexes that you should create from that interface, you manually had to click each of the 10 advised indexes to create the index. The i5/OS V6R1 Index Advisor interface allows you to accomplish this task more quickly by simply highlighting a group of advised indexes and selecting the **Show SQL** task to generate all the Create Index statements with a single operation.

Further simplifying the DB2 index advice process is a capability known as the *Index Advice Condenser*. The Condenser can also be leveraged on i5/OS V5R4 by installing the latest Database Group PTF and iSeries Access for Windows fixes for that release.

This new Condenser simplifies the use of DB2 Index Advisor by analyzing the suggested indexes and trying to combine overlapping index advice into a single index recommendation. Overlapping index advice can occur when the query optimizer recommends indexes with the key columns in a different order or recommends indexes for different SQL requests that share some of the key columns. For example, a series of queries and SQL reports that run against an inventory table can cause the DB2 Index Advisor to recommend two indexes with the following keys:

- Index #1 — Key columns: item_color
- Index #2 — Key columns: item_color, item_size

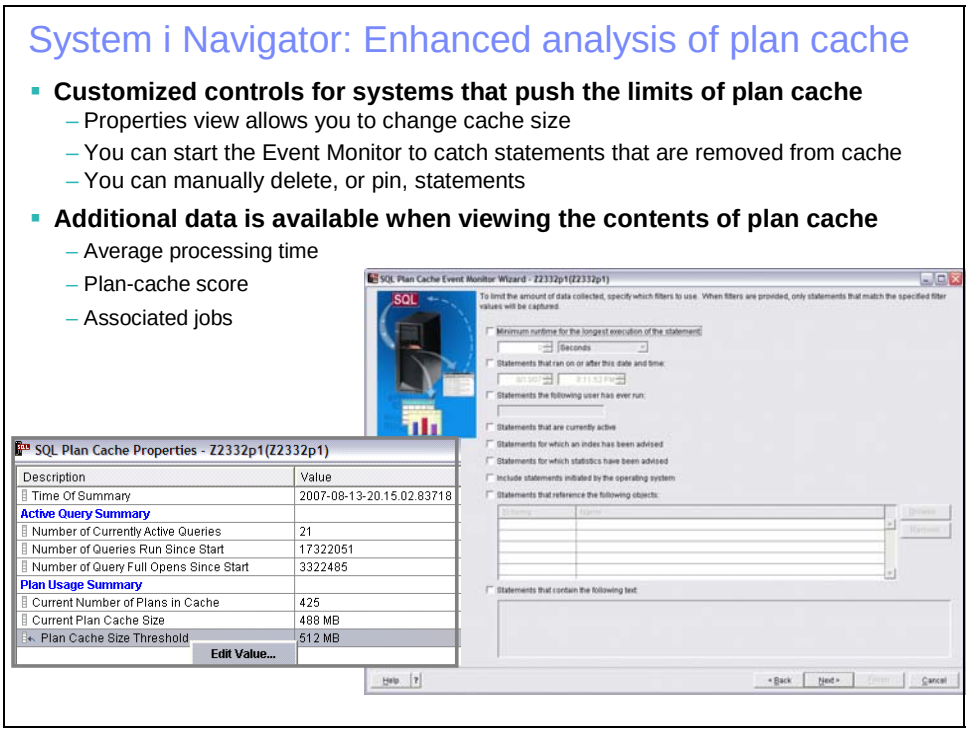
In this case, creating a single index with key columns of item_color and item_size covers both index recommendations. You do not have to manually correlate and combine advice for similar indexes because the Index Advice Condenser automates this process.



System i Navigator: Enhanced Index Evaluator

i5/OS V6R1 also simplifies the index-evaluation capabilities of the Show Indexes task when a group of DB2 objects is involved. The Show Indexes task has been a favorite of users since it was introduced in i5/OS V5R3 — because this tool allows you to easily evaluate which indexes and logical files DB2 for i5/OS uses and which index and LF objects are wasting the disk space and system resources that are used to maintain them. The drawback with this tool is that you can only run it at a table level. When it is necessary to perform index evaluation for all 50 tables (physical file) in a schema (library), you have to launch the Show Indexes task for each table level. Luckily, that is no longer an issue with DB2 for i5/OS V6R1, because you can run the Show Indexes task at a schema level by right-clicking the Tables object in a schema and then selecting the task (as shown in this screen capture).

This index evaluation interface also allows you to reset the usage statistics for an index. This can be helpful when you need to determine how often an index is used in the newly tuned environment.



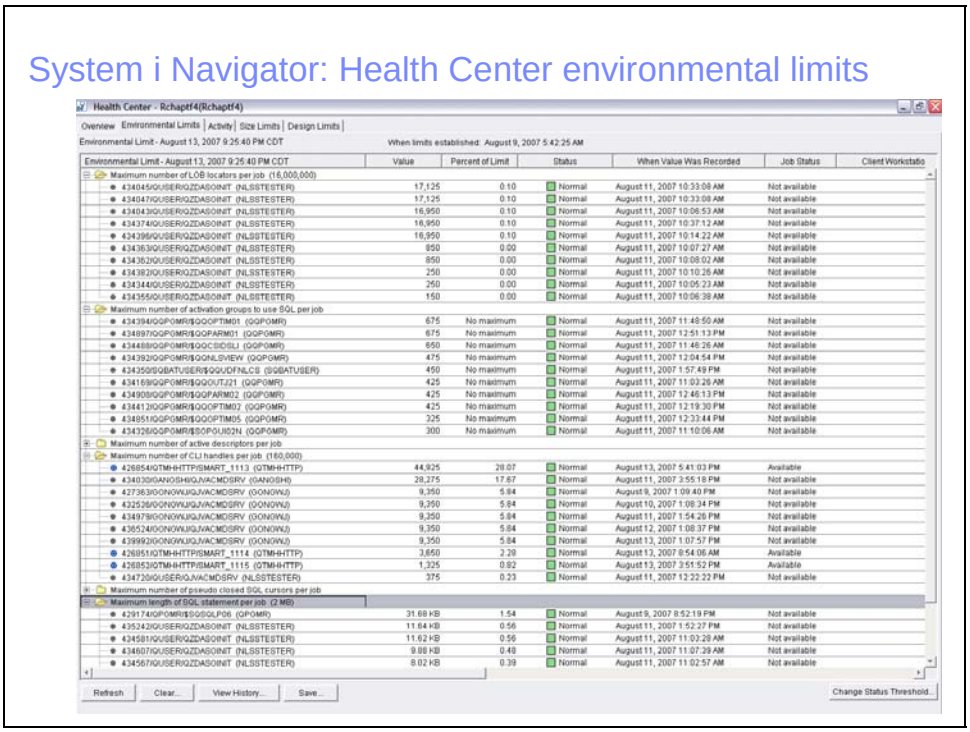
System i Navigator: Enhanced analysis of plan cache

The SQL plan cache is an internal repository that the SQE uses to store the access plans and associated statistics for SQL statements that are currently running or have recently run on your server. The i5/OS V5R4 plan-cache analysis tools are enhanced in DB2 for i5/OS V6R1 to make plan-cache analysis even simpler.

There are additional metrics for each plan-cache entry — to help you better understand the historical usage of a plan, such as the average processing time, the jobs that use an access plan, and a plan cache's score. The score for a plan-cache entry describes how likely it is for an access plan to remain in the cache when the maximum plan-cache size limit is exceeded.

Administrators can also change a plan's score to a higher value, giving it more immunity when the plan cache needs to be pruned to remain under the maximum-size limit. You can also manually delete plans from the plan cache to avoid exceeding the size-limit threshold. Furthermore, you can increase the maximum size of the plan cache from the Plan Cache properties dialog, as shown on the bottom-left corner of this chart.

The window that is displayed on the right side of this chart is the interface for the new SQL Plan Cache Event Monitor, which you can use to catch those SQL statements whose plans are removed from the plan cache by DB2. You can analyze the Plan Cache Event Monitor data later to determine if the SQL statements that were removed are critical and must, therefore, remain in the plan cache. If an SQL statement needs to remain in the plan cache, you can either increase the size of the plan cache or increase the plan-cache score for that SQL statement.



System i Navigator: Health Center environmental limits

The final System i Navigator enhancement to discuss is the i5/OS V6R1 modifications to the Health Center. The Health Center is enhanced to report on DB2 runtime limits, such as the number of large-object (LOB) locators or CLI handles per connection or job.

Summary

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

Summary

As you have just learned, DB2 for i5/OS V6R1 delivers a lot of improvements that support your application-development efforts. These enhancements make it easier for you to integrate your SQL and RPG code, connect to Java and .NET environments, troubleshoot performance issues and much more.

Spend a few moments reviewing the Web sites provided in the References section of this course. These sites will help you gain an even deeper understanding of the value that you can derive by working with DB2 for i5/OS V6R1.

Resources

- **DB2 for i5/OS Web sites**
  - Home Page
    ibm.com/systemi/db2
  - IBM developerWorks® Zone
    ibm.com/developerworks/db2/products/db2i5OS
  - IBM Porting Zone
    ibm.com/servers/enable/site/db2/porting.html
- **Newsgroups**
  - USENET
    www.comp.sys.ibm.as400.misc, comp.databases.ibm-db2
  - System i Network DB2 Forum
    http://systeminetwork.com/isnetforums/forumdisplay.php
- **Education resources (classroom and online)**
  - **ibm.com**/systemi/db2/gettingstarted.html
- **DB2 for i5/OS publications**
  - White papers
    ibm.com/partnerworld/wps/whitepaper/i5os
  - Online manuals
    ibm.com/systemi/db2/books.html
  - IBM Redbooks®
    ibm.com/redbooks
    - *Getting Started with DB2 Web Query for System i* (SG24-7214)
      www.redbooks.ibm.com/abstracts/sg247214.html?Open
    - *OnDemand SQL Performance Analysis Simplified on DB2 for i5/OS in V5R4* (SG24-7326)
      www.redbooks.ibm.com/abstracts/sg247326.html?Open
    - *Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS* (SG24-6598)
      www.redbooks.ibm.com/abstracts/sg246598.html?Open
    - *Modernizing iSeries Application Data Access* (SG24-6393)
      www.redbooks.ibm.com/abstracts/sg246393.html?Open
- **IBM i5/OS Information Center**
  http://publib.boulder.ibm.com/iseries

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

AS/400, DB2, developerWorks, i5/OS, IBM, the IBM logo, Integrated Language Environment, OS/400, PartnerWorld, Rational, Redbooks, System i and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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.

Intel, Intel Inside (logos), MMX and Pentium 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.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Other company, product, or 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 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.

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.