DB2 Version 9.1 for z/OS

What's New?

GC18-9856-00

DB2 Version 9.1 for z/OS

What's New?

GC18-9856-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 37.

First Edition (March 2007)

This edition applies to DB2 Version 9.1 for z/OS (DB2 V9.1 for z/OS), product number 5635-DB2, and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

© Copyright International Business Machines Corporation 2007. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book v
Terminology and citations v
How to send your comments v

Chapter 1. Introduction 1

Chapter 2. Leveraging your enterprise for information on demand 3
Seamless integration of XML data and relational data 3
Compatibility and leadership with SQL 4
Tools that support your enterprise 15

Chapter 3. Availability 21
Online REORG with no BUILD2 phase 21
Faster replacement of one table with another 21
Universal table spaces 21
Better availability during REBUILD INDEX operations 22
Improved availability with column and index renaming capabilities 23
Modify EARLY code without an IPL 23
ALTER TABLESPACE and index logging improvements 23
Support for using SMS storage classes with DB2-defined data sets 23

Chapter 4. Performance 25
Reduction in CPU processing time for utilities 25
SQL optimization improvements 25
Indexing improvements 29
Improved performance for varying-length rows 30
Relief for sequential key insert 30
Improved logging performance 30
Improved data insert performance 31

Chapter 5. Regulatory compliance 33
Roles and network trusted contexts 33
Improved auditing 33
Support for Secure Socket Layer protocol 33
More security options with INSTEAD OF triggers 34

Appendix. How to obtain DB2 information 35
DB2 on the Web 35
DB2 publications 35
DB2 education 36
How to order the DB2 library 36

Notices 37
Trademarks 38

Glossary 39

Information resources for DB2 for z/OS and related products 43

Index 49

© Copyright IBM Corp. 2007

About this book

This book provides an executive overview of new function in DB2 Version 9.1 for z/OS. The topics in this book provide a framework for describing new function in DB2® for z/OS®. New functions are categorized according to user benefits such as information on demand, availability, and performance.

This book is written primarily for people who are evaluating DB2.

Terminology and citations

In this information, DB2 Version 9.1 for z/OS is referred to as "DB2 for z/OS" for the general case or as "V9.1" when different versions of DB2 for z/OS are discussed. In cases where the context makes the meaning clear, DB2 for z/OS is referred to as "DB2." When this information refers to titles of DB2 for z/OS books, a short title is used. (For example, "See *DB2 SQL Reference*" is a citation to *IBM® DB2 Version 9.1 for z/OS SQL Reference*.)

When referring to a DB2 product other than DB2 for z/OS, this information uses the product's full name to avoid ambiguity.

The following terms are used as indicated:

DB2 Represents either the DB2 licensed program or a particular DB2 subsystem.

OMEGAMON
Refers to any of the following products:
- IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS
- IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS
- IBM DB2 Performance Expert for Multiplatforms and Workgroups
- IBM DB2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS® Represents CICS Transaction Server for z/OS.

IMS™ Represents the IMS Database Manager or IMS Transaction Manager.

MVS™ Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF®
Represents the functions that are provided by the RACF component of the z/OS Security Server.

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 for z/OS documentation. You can use the following methods to provide comments:

- Send your comments by e-mail to db2zinfo@us.ibm.com and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title or a help topic title).

© Copyright IBM Corp. 2007

- You can send comments from the Web. Visit the library Web site at:

  www.ibm.com/software/db2zos/library.html

  This Web site has an online reader comment form that you can use to send comments.
- You can also send comments by using the feedback link at the footer of each page in the Information Management Software for z/OS Solutions Information Center at http://publib.boulder.ibm.com/infocenter/db2zhelp.

Chapter 1. Introduction

DB2 Version 9.1 for z/OS (V9.1) delivers exciting new innovations, in key areas, that help you realize the business value of your company's information.

Read on for a brief overview of the innovations that are delivered in V9.1.

Decreased total cost of ownership

The total cost of ownership for maintaining an information management infrastructure is determined by many factors, including the costs of hardware and software and the number of staff required.

With V9.1, reductions in your total cost of ownership can be realized through savings in hardware and software, and productivity for people. The productivity comes through improved SQL, the addition of XML, and the reduction in database administration tasks. System z9™ offerings such as zIIP and new Business Class and Enterprise Class Processors are economical and provide extra capacity and reductions in DB2 costs for some distributed, parallel query, and utility index workloads. To ease the challenging tasks that face database administrators, some tasks have been automated or eliminated. As a result, a database administrator can manage more terabytes of data and take on more business tasks.

More power to leverage enterprise for information on demand

V9.1 delivers a number of capabilities that help you leverage your enterprise for information on demand through increased programmer productivity, increased compatibility in the DB2 family of products, and improvements for managing enterprise data.

Native support for pureXML increases productivity of programmers and provides unprecedented scalability and performance. With V9.1 at the core of a service-oriented architecture, you can have seamless integration of XML and relational data. Enhancements to SQL increase family compatibility and leadership. Improvements and additions in the QMF™ family of products improve access to your enterprise data. New and enhanced DB2 Tools improve management of your enterprise data. Most of these improvements also provide benefits for key enterprise application partners (for example SAP, PeopleSoft, and Siebel) and their customers.

Increased availability through database definition on demand

Version 8 delivered online schema change capabilities that increased database availability with less disruption.

V9.1 expands those capabilities to deliver data definition on demand. For example, improvements in online reorganization of table spaces for a few partitions is improved greatly. One of the important changes is the ability to replace one table with another table quickly. Another important change is the ability to rename a column or an index.Other capabilities let you modify, alter, create, and rebuild objects and code either online or by other means that are less disruptive to the availability of your data.

© Copyright IBM Corp. 2007

Better performance

Complex applications include both transactions and reporting. Performing both transactions and reporting well is imperative to running an enterprise efficiently. V9.1 delivers a number of enhancements that boost transaction performance and help reduce your total cost of ownership through reduced CPU time, improved query performance, faster disk access, and improved logging and insert performance.

Reductions in CPU time are observed for LOAD and REORG utilities and varying-length data. In addition, queries that are written in the native SQL procedural language are eligible to run on zIIP. Also, several improvements in disk access can reduce the time for sequential disk access.

Key improvements for reporting include optimization enhancements that improve query and reporting performance and ease of use. Improved query performance SQL improvements include more queries, more consistency with the database management industry and within the DB2 family (which contributes to ease of use and the ability to port applications to DB2 for z/OS).

Regulatory compliance

Regulatory compliance, security and auditing are in the headlines and growing much more important. V9.1 helps you respond to increasing needs to protect security, to assure integrity, and to comply with regulations such as the Sarbanes-Oxley Act, the Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry (PCI) Security Standards. Improved access control with network trusted context and roles allows more precise control of security. Improved filtering makes auditing more usable. Secure Socket Layer (SSL) data encryption on networks is more secure.

Increased synergy with System z™

As in Version 8, V9.1 takes advantage of the latest improvements in System z9 hardware and software to provide better performance, improved value, more resilience, and better function.

For example, remote native SQL procedures are now enabled for processing on the IBM System z9 Integrated Information Processor (zIIP). DB2 synergy with System z9 continues with a range of I/O improvements in channels, in disks, and in DB2. Tests with Version 8 that involve parallel access on one channel have shown faster response times for sequential access; similar improvements are anticipated for V9.1. For example, one early performance test for sequential performance achieved 183 MB per second. V9.1 also takes advantage of new System z9 hardware support with a new decimal floating-point data type that lets you use decimal floating-point numbers with greater precision. Flashcopy can be used for DB2 database backup and restore operations. Other improvements for V9.1 on System z9 include added security and encryption and Unicode collation.

Chapter 2. Leveraging your enterprise for information on demand

DB2 Version 9.1 for z/OS delivers functions and support that increase compatibility within the DB2 family and let you more easily leverage your enterprise for information on demand.

Increases in family compatibility translate to increases in productivity for programmers who work in a standard application environment and increases in the portability of your applications. Enhancements that address these areas include pureXML support in the DB2 database and compatibility and family leadership through various new and improved SQL functions and statements.

In addition to enhancing programmer productivity and application portability, improvements in DB2 QMF, and an array of DB2 Tools and new tools offerings such as the Optimization Service Center and DB2 Accessories Suite, let you access your data and manage your enterprise like never before.

Most of the new and improved functions deliver benefits for both business partners and customers.

Seamless integration of XML data and relational data

XML is an important innovation that enables business-to-business communication of data, regardless of differences in the systems that receive and work with the data.

With XML, you can easily send semi-structured data across the Web without losing the relationship between data within a document. Because the content of an XML document is independent from the formatting instructions, data can be rendered appropriately in a variety of output formats. Rich support of XML within a database management system provides advantages that include more efficient storage, query, and indexing capabilities.

Support for XML capabilities and functions span the entire DB2 family. Version 8 of DB2 for z/OS and Version 8 of DB2 for Linux®, UNIX® and Windows® provide basic support for storing, retrieving, and querying XML documents. DB2 9 for Linux, UNIX and Windows continues the work by delivering rich support of XML, including an XQuery interface to the data. Now, DB2 V9.1 for z/OS expands on similar support by delivering seamless integration of XML data and relational data in the DB2 database. Functional changes are provided in the following areas:

- XML document storage
- XML document retrieval
- Application development
- Database administration support
- Peformance benefits through indexing support

XML document storage

DB2 V9.1 for z/OS provides fully integrated storage of XML data in the DB2 database system, which lets your client applications access and manage the XML data by leveraging DB2 functionality. The XML column data type is provided for

© Copyright IBM Corp. 2007

storing XML data in DB2 tables. Most SQL statements support the XML data type. As a result, you can perform many common database operations with XML data, such as creating tables with XML columns, adding XML columns to existing tables, creating indexes over XML columns, creating triggers on tables with XML columns, and inserting, updating, or deleting XML documents. A decomposition stored procedure is also provided. With this stored procedure, you can extract data items from an XML document and store those data items in columns of relational tables.

XML document retrieval

You can use SQL to retrieve entire documents from XML columns in a way that is similar to retrieving data from any other type of column. When you need to retrieve portions of documents, you can specify XPath expressions, through SQL with XML extensions (SQL/XML).

Application development

Application development support of XML in DB2 V9.1 for z/OS enables applications to combine XML, relational data access, and storage. The following programming languages support the new XML data type:

- Assembler
- C or C++ (embedded SQL or DB2 CLI)
- COBOL
- Java™ (JDBC or SQLJ)
- PL/I

Database administration support

Database administration support includes:

- An XML schema repository for all XML schemas that are required to validate and process XML documents that are stored in XML columns or decomposed into relational tables
- The ability to use DB2 for z/OS utilities to perform operations on XML objects in a way that is similar to handling LOB objects

Performance benefits through indexing support

V9.1 offers performance benefits through indexing support, which is available for data that is stored in XML columns. Using indexes over XML data can improve the efficiency of queries that you issue against XML documents.

Compatibility and leadership with SQL

V9.1 delivers functional changes in SQL that increase productivity of programmers through family compatibility and leadership.

Version 8 of DB2 for z/OS took big steps toward improving consistency of SQL across the DB2 family by providing more functions that are common between DB2 for Linux, UNIX and Windows. V9.1 and DB2 9 for Linux, UNIX and Windows moves even more SQL function from the unique set to the common set. With the increase in common SQL, programmer productivity also increases. V9.1 also introduces several new SQL functions that are firsts in the DB2 family.

SQL consistency improvements

Increased consistency with SQL across the DB2 family is good news for programmers who write cross-platform applications, because porting and writing application programs becomes easier.

The following SQL functions, statements, and clauses are consistent across the DB2 family.

SELECT FROM UPDATE or SELECT FROM DELETE function

The insert within select feature that was introduced in Version 8 of DB2 for z/OS has been expanded in V9.1 to support the retrieval of columns from rows that are modified by a SELECT FROM DELETE or a SELECT FROM UPDATE statement. Now, one SQL call to DB2 modifies the table contents and returns the resultant changes to the application program. In addition, an application can now be coded to perform a destructive read from a table when a SELECT FROM DELETE statement is included. This feature is particularly useful when a table is used as a data queue.

INSTEAD OF triggers

In previous versions of DB2 for z/OS, operations that required different rules for read and write access (for example, encryption and decryption) required users to build into their applications an awareness that the object for read access is a view, and that the object for write access is a base table. V9.1 reduces the complexity in your application programs by introducing INSTEAD OF triggers.

INSTEAD OF triggers are defined on views only, and are used to process insert, update, and delete operations (through trigger logic) rather than relying on the INSERT, UPDATE, or DELETE statement to activate the trigger. With the INSTEAD OF trigger, an application doesn't need to include the complexity that specifies which operations are performed against views and which operations are performed against the base table because the activated trigger makes the operations seem as if they are performed against a view.

Examples of using INSTEAD OF triggers include updates through join queries or encoding and decoding data from the database within a view. In the latter case, the view might encapsulate decryption functions while the INSTEAD OF triggers use the encryption functions to ensure security within the database.

BIGINT data type and function

V9.1 introduces a new SQL data type and function, BIGINT (big integer), that supports big integers and extends the set of currently supported exact numeric data types (SMALLINT and INTEGER).

A big integer is a binary integer that has a precision of 63 bits. The BIGINT data type can represent 63-bit integers, and is compatible with all numeric data types. The BIGINT function returns a big integer representation of a number or a string representation of a number. You can store or load big integer values into DB2 tables, and manipulate the data in a variety of ways. In addition, you can use the CAST specification to increase the portability of your applications.

BINARY data type and function

V9.1 provides expanded support for binary string data by introducing the BINARY data type and function. The VARBINARY (varying-length binary string) data type and function are also introduced. The VARBINARY data type and function are not yet supported by other members of the DB2 family.

The BINARY data type represents a fixed-length binary string. When fixed-length binary string distinct types, columns, and variables are defined, the length attribute is specified, and all values have the same length. A binary string column is used for storing non-character data, such as encoded or compressed data, pictures, voice, and mixed media. A binary string column can also hold structured data for use by distinct types, user-defined functions, and stored procedures. The BINARY function returns a BINARY (fixed-length binary string) representation of a string of any type or a row ID type.

Related concepts

"VARBINARY data type" on page 13

File reference variables

A *file reference variable* is a host variable that is defined in a host language (for example C or COBOL) to contain the file name that directs file input and output for a large object (LOB). V9.1 adds support for two new file reference variables, LOB and XML.

DB2 support for file reference variables allows a large LOB or an XML value to be inserted from a file or selected into a file rather than a host variable; the application does not need to acquire storage to contain the LOB or XML value. File reference variables also facilitate the movement of LOB or XML values from the database server to a client application or from a client application to a database server without going through the working storage of the client application. In addition, file reference variables bypass host language limitations on the maximum allowed size for LOB values that are located in working storage. The following host languages support file reference variables:

- C and C++
- COBOL
- PL/I
- Assembler
- REXX

INTERSECT keyword in a fullselect

The new INTERSECT keyword specifies the set operator, intersection, in a fullselect operation between two result tables.

Two types of INTERSECT operations are available: INTERSECT ALL and INTERSECT DISTINCT. If you specify INTERSECT ALL, the result consists of all rows that are in both the first result table and the second result table, and redundant duplicate rows are included. If you specify INTERSECT DISTINCT, the result consists of all rows that are in the first and second result tables, and the redundant duplicate rows are eliminated. In either case, each row of the result exists in the first and second result tables.

EXCEPT keyword in a subselect

The new EXCEPT keyword specifies the set operator, difference, in a fullselect operation between two result tables.

Two types of EXCEPT operations are available: EXCEPT ALL and EXCEPT DISTINCT. If you specify EXCEPT ALL, the result consists of all rows that are only in the first result table, and redundant duplicate rows are included. If you specify EXCEPT DISTINCT, the result consists of all rows that are only in the first result table, and the redundant rows are eliminated. In either case, each row in the result is a row from the first result table that does not have a matching row in the second result table.

Native support for SQL procedures

In V9.1, new support for native SQL procedures simplifies the definition and use of SQL procedures by eliminating the need for generating a C program.

In previous versions of DB2 for z/OS, a C program must be generated from transformed SQL statements and SQL control statements that are contained in the SQL procedure. The resulting C program is then run like an external stored procedure. Now, when you create a native SQL procedure in V9.1 new-function mode, the procedural statements are converted to a representation that is stored in the database directory, as is true for other SQL statements. The parameter list procedure options are stored in the database catalog tables as in prior releases. When a native SQL procedure is called, the representation is loaded from the directory, and the DB2 engine runs the procedure. A number of additional functions and extensions in V9.1 provide consistency with the SQL Standard and with the rest of the DB2 family. Examples of these improvements include support for:

- Changing name resolution within a procedure body
- Using delimited identifiers, including lowercase characters, for SQL condition names, SQL labels, SQL variables, and SQL parameters
- Nested compound statements, including a compound statement within the body of a condition handler
- Versioning and managing source code
- Deploying of native SQL procedures to multiple servers
- Debugging of native SQL procedures

Nested compound statements in SQL procedures

SQL procedures now support nested compound statements.

Compound statements introduce a block of SQL statements in an SQL procedure. Before V9.1, the body of an SQL procedure could contain a single compound statement (which could contain other SQL statements, but not another compound statement), or a single SQL procedure statement other than a compound statement. As a result, a condition handler could not contain a compound statement, either.

Now, with support for nested compound statements in SQL procedures, you can:

- Use a compound statement within a condition handler
- Use nested compound statements to define different scopes for SQL variables, cursors, condition names, and condition handlers.

Expanded support for not logging table spaces

Version 8 of DB2 for z/OS provides support for suspension of logging in global temporary tables, LOB table spaces, during the LOAD REPLACE operation, and during an insert of a declared temporary table. Now, V9.1 supports suspension of logging for base table spaces, XML table spaces, and the indexes that are associated with them. You can specify the logging attributes, LOGGED or NOT LOGGED, at the table space level when you alter or create tables and table spaces, and when you create tables or auxiliary tables. You can also suppress logging during the online LOAD RESUME process.

Because the existing logging facilities in DB2 are already finely tuned, suspension of logging does not generally improve the performance of your system. However, the ability to suspend record logging is useful in a variety of situations in which data is being duplicated and loss of concurrency and recoverability is not a concern. In those cases, if the data is lost, you can re-create or regenerate it from

the original source instead of using an image copy and applying log records. Examples of tables for which suspension of logging is advantageous include materialized query tables, summary tables, tables to which data is propagated, and temporary tables that are populated with the result set from a query as an intermediate step in an application. An additional benefit of suspended logging is improved scalability, particularly for operations that insert large volumes of data with the INSERT statement.

OLAP specifications for RANK, DENSE_RANK, and ROW_NUMBER

Online analytical processing (OLAP) specifications provide the ability to return ranking, row numbering, and existing aggregate function information as a scalar value in the result of a query.

You can include OLAP specifications in an expression, in a select-list, or in the ORDER BY clause of a select-statement. The result to which the OLAP specification is applied is the result table of the innermost subselect that includes the OLAP specification. V9.1 delivers OLAP specifications for RANK, DENSE_RANK, and ROW_NUMBER.

RANK and DENSE_RANK

RANK and DENSE_RANK specify that the ordinal rank of a row within the specified window is computed. Rows that are not distinct with respect to the ordering within the specified window are assigned the same rank. You can define the results of ranking with gaps in the sequential rank numbering by using the RANK specification, or without gaps, by using the DENSE_RANK specification. Common examples of using RANK or DENSE_RANK specifications include:

- Ranking sales figures (for example, determining which stores in a chain have the lowest and highest sales)
- Ranking employees within a department or division according to various indicators
- Creating "top-*n*" queries (for example, retrieve the five employees that have the highest salaries)

ROW_NUMBER

ROW_NUMBER specifies that a sequential row number is computed for the row that is defined by the ordering, starting with 1 for the first row. If the ORDER BY clause is not specified in the window, the row numbers are assigned to the rows in an arbitrary order, as the rows are returned (but not according to any ORDER BY clause in the select-statement). You can use ROW_NUMBER to number the result rows of a query. Row numbers also enable easy formulation of queries for computing histogram statistics (quantile computations), and they enable formation of other OLAP specifications (for example, moving sums, moving averages, and so on).

COLLATION_KEY function

The new COLLATION_KEY function supports culturally correct and case-insensitive collation of Unicode data.

The COLLATION_KEY function processes a Unicode UTF-16 input string and a collation name, and returns a varying-length binary sort key. The result of the COLLATION_KEY operation can be compared to the result of another COLLATION_KEY operation on another string to determine their order within the

specified collation name. The attributes of the collation name specify the collation characteristics (for example locale attribute, treatment of accent and case, an so on) for the sort key, and conform to the conventions of the z/OS-supported Unicode conversion services.

You can also use the COLLATION_KEY function to perform case-insensitive (or *caseless*) comparisons of string expressions by specifying attributes on the collation name that either ignore case or ignore a combination of case and other attributes such as spaces, punctuation, and symbols.

Capability to create an index on an expression

In V9.1, an extension to the CREATE INDEX statement lets you create an index on an expression (also known as an *index on expression*).

An expression can be a column reference, a built-in function invocation, or a general expression with certain restrictions. Unlike a simple index, the index key of an index on expression is composed by concatenating the result (also known as a key-target) of the expression that is specified in the ON clause. An index that is created on an expression lets a query take advantage of index access (if the index is chosen by the optimizer) and avoid a table space scan.

Automatic creation of a database, a table space, and all system-required objects

The function of the CREATE TABLE statement now supports automatic (implicit) creation of a database or a table space.

Implicit creation of a database, a table space, and all system-required objects: A database, a table space, and all system-required objects are created implicitly if you do not name a table space or a database in the IN clause of a CREATE statement, and the CREATE IMPLICIT DATABASES installation parameter is set to YES. The attributes of the table space (for example, table space type, the underlying data sets, and use of data compression) are determined by table space installation parameters. If a table space is implicitly created, the following system-required objects are also created:

- The enforcing primary key index
- The enforcing unique key index
- The ROWID index (if the ROWID column is defined as GENERATED BY DEFAULT)
- A LOB table space, auxiliary table, and auxiliary index

Implicit creation of a table space: A segmented table space is created implicitly in DSNDB04 if you do not name a table space or a database in the IN clause of a CREATE statement, and the CREATE IMPLICIT DATABASES installation parameter is set to NO.

IBM Spatial Support for DB2 for z/OS

The new IBM Spatial Support for DB2 for z/OS allows you to generate and analyze spatial information about geographic features and to store and manage the data on which this information is based.

Spatial Support for DB2 for z/OS provides of a set of spatial data types, user defined functions, and stored procedures for spatial related queries. With Spatial Support, you can:

- Invoke spatial queries for local and remote clients to answer questions based on geographic relationships.

- Create spatial indexes on spatial columns, which can improve query performance when you use spatial predicate functions.
- Manage geographic coordinate systems, spatial indexes, and spatial column usages through stored procedure interfaces.

In addition, Spatial Support includes an ODBC program that you can use to enable and disable the spatial feature. This program can invoke each of the stored procedures through a set of command line arguments.

Leverage existing application programming skills

Increased consistency across the DB2 family lets you leverage the existing skills of your application programmers through support of a wider range of tools, environments, and languages. V9.1 improves support for end-to-end application development by connecting new languages and environments to the scale and value of the existing infrastructure. The IBM Rational® and WebSphere® product lines provide part of the connection, with products like Rational Data Architect, Rational Application Developer, and WebSphere Information Integration. DB2 clients provide more support for new environments and new languages.

V9.1 supports key database technologies, including SQL and native SQL procedures, XML, service-oriented architecture (SOA), and various Web services.

A wide variety of development frameworks provide language support and application development and deployment tools for Web-based applications that work with the DB2 family. These frameworks also support both traditional programming languages and newer, open source languages. For example, development communities exist for the following traditional programming languages:
- COBOL, PL/I, REXX, C, C++, Fortran, assembler, and APL2®
- Java support, including JDBC and SQLJ
- Microsoft® language products, including .NET, Visual C# for .NET, and Visual Basic for .NET

Examples of Open Source development support include:
- Eclipse
- PHP and Zend Core for IBM
- Perl
- Python
- Ruby on Rails
- TOAD for DB2

Enhancements to large object support

V9.1 offers many improvements for handling large objects (LOBs) through extensions to SQL statements and enhancements to utilities and performance.

SQL enhancements for large objects

V9.1 provides several SQL enhancements for large objects:
- Enhancements to the FETCH statement for LOB and XML data
- Support for LOB file reference variables

Enhancements to the FETCH statement for LOB and XML data

In previous versions of DB2, applications that worked with LOBs had two primary methods for fetching LOB data: Fetching data into a pre-allocated buffer and using a LOB locator to retrieve a handle on the data. Fetching data into a preallocated

buffer has the potential for causing virtual storage constraint problems, especially LOBs that have a large maximum length. In addition, an application that uses LOB locators that commit infrequently or do not explicitly free the locators can use considerable amounts of DB2 resources. The introduction of XML objects in V9.1 presents additional complications because XML columns do not have locators, and when a table containing an XML column is created, no maximum length is defined for the XML object.

V9.1 introduces the WITH CONTINUE clause on the FETCH statement to address the problems that are encountered when an application fetches LOB and XML data. You can use the WITH CONTINUE clause in an application to retrieve LOB and XML columns in multiple pieces without using a LOB locator, and continue a FETCH operation to retrieve the remaining data for LOB and XML columns when truncation occurs. The application manages the buffers and reassembles the pieces of data. Two common uses for FETCH CONTINUE include:

- Dynamic allocation of appropriate storage size: You can use the initial FETCH statement to fetch data into a preallocated buffer of a moderate size. If the returned data item is too large to fit in that buffer, you can use the length information that is returned to allocate the right amount of storage, and use one FETCH CONTINUE statement to retrieve the remaining data.
- Streaming data through a single, fixed-size buffer: If more data remains after a FETCH operation, you can use as many FETCH CONTINUE statements as necessary to retrieve the data and use the same buffer area. In this case, the data in the buffer must be processed after each FETCH or FETCH CONTINUE operation.

File reference variables for large objects

A *file reference variable* is a host variable that is defined in a host language (for example C or COBOL) to contain the file name that directs file input and output for a large object (LOB). V9.1 adds support for a LOB file reference variable.

With DB2 support for file reference variables, large LOB values can be inserted from a file or selected into a file rather than a host variable; the application does not need to acquire storage to contain the LOB value. File reference variables also facilitate the movement of LOB values from the database server to a client application or from a client application to a database server without going through the working storage of the client application. In addition, file reference variables bypass host language limitations on the maximum allowed size for LOB values that are located in working storage. The following host languages support file reference variables:

- C and C++
- COBOL
- PL/I
- Assembler
- REXX

Utilities enhancements for large objects

V9.1 provides enhancements for handling large objects (LOBs) in the following utilities: LOAD, UNLOAD, REORG, CHECK LOB, and CHECK DATA.

Improved LOB handling for LOAD and UNLOAD utilities

Because of enhancements to the LOAD utility, an input field value can contain the name of the file that contains a LOB column value. The LOB column value is loaded from that file.

Enhancements to the UNLOAD utility let you store the value of a LOB column in a file and record the name of the file in the unloaded record in the base table.

Improved REORG of LOB table spaces

Before V9.1, you cannot access LOB data during the REORG operation, and LOBs are moved within the existing LOB table space, which prevents physical space from being reclaimed from the LOB data set. Improvements in V9.1 overcome those previous limitations. Now, the original LOB table space is drained of writers. All LOBs are then extracted from the original data set and inserted into a shadow data set. When this operation is complete, all access to the LOB table space is stopped (the readers are drained) while the original data set is switched with the shadow data set. At this point, full access to the new data set is enabled, and an inline copy is taken to ensure recoverability of data.

CHECK LOB and CHECK DATA utilities

The options SHRLEVEL REFERENCE and SHRLEVEL CHANGE have been added to both CHECK LOB and CHECK DATA utilities. These options dramatically reduce the amount of time during which the data is in the read-only state when you run CHECK LOB and CHECK DATA utilities.

The CHECK DATA utility checks table spaces for violations of referential and table check constraints and reports information about violations that it detects. CHECK DATA also checks for consistency between a base table space and the corresponding LOB or XML table spaces. The new SHRLEVEL REFERENCE and SHRLEVEL CHANGE options provide the following functions:

- CHECK DATA with SHRLEVEL REFERENCE specifies that applications can read from but cannot write to the index, table space, or partition that is to be checked during CHECK DATA processing.
- CHECK DATA with SHRLEVEL CHANGE specifies that applications can read from and write to the index, table space, or partition that is to be checked during CHECK DATA processing.

The CHECK LOB utility identifies any structural defects in the LOB table space and any invalid LOB values. You run the CHECK LOB utility before the CHECK DATA utility if a table space contains at least one LOB column. The new SHRLEVEL REFERENCE and SHRLEVEL CHANGE options provide the following functions:

- CHECK LOB with SHRLEVEL REFERENCE specifies that applications can read from but cannot write to the table space that is to be checked during CHECK LOB processing.
- CHECK LOB with SHRLEVEL CHANGE specifies that applications can read from and write to the table space that is to be checked during CHECK LOB processing.

Performance enhancements for large objects

Several enhancements improve performance for retrieving data for small and medium LOBs and eliminate locks for LOB operations.

Optimization of data retrieval for small and medium LOBs

In prior releases, processing large LOBs, particularly from a distributed perspective, has been optimized for retrieving larger amounts of data. Many applications effectively use locators to retrieve LOB data regardless of the size of

the data that is retrieved. The mechanism involved incurs a separate network flow to determine the length of the data to be returned. Locators in this process remain active for the scope of the transaction, which can be longer than necessary for smaller amounts of data unless the locators are explicitly freed. As a result, valuable server resources are consumed.

V9.1 introduces the ability for a server to dynamically determine the most efficient mode in which to return LOB or XML data. When the dynamic data format is enabled, the life span of the locator is the scope of the cursor (a cursor-based locator) instead of the scope of the transaction. A new mechanism is also provided to allow the requester to retrieve sequential chunks of the LOB data while maintaining the position of the data (by means of the locator) at the server.

Increased performance through elimination of LOB locks

LOB locks are used to serialize LOB table space access and to determine whether the previously deallocated LOB space can be reallocated. In prior releases, DB2 acquires a lock on the LOB value while performing insert, update, delete, and select operations, and during LOB space allocation. V9.1 eliminates the acquisition of locks on every LOB operation including LOB locks that are used for space allocation. The requirement for a LOB lock is removed for insert, delete, update, and select operations. In addition, a LOB lock is no longer required to serialize the consistency between the value of the LOB and the column of the base row for an uncommitted read operation. As a result, lock escalation for LOB locks at the table space level is also eliminated. This enhancement improves the overall elapsed time for LOB data retrieval.

SQL leadership: family firsts

V9.1 provides leadership in SQL by offering a number of DB2 family firsts.

TRUNCATE TABLE statement

The TRUNCATE TABLE statement provides an efficient mechanism for deleting all data rows in a designated DB2 table without activating delete triggers or altering the current table attributes in the DB2 catalog. The statement also provides an IMMEDIATE option to permanently empty the designated DB2 table without issuing a commit, and it provides a REUSE STORAGE option to allow reuse of deallocated storage.

DECFLOAT built-in data type

DECFLOAT (decimal floating point) is a new built-in SQL data type that has a maximum precision of 34 digits. DECFLOAT data can be manipulated, stored, or loaded into DB2 tables.

A decimal floating-point value is an IEEE 754r (finite) number with a decimal point. The position of the decimal point is stored in each decimal floating-point value.

VARBINARY data type

V9.1 provides support for the VARBINARY data type and function. The VARBINARY data type represents a varying-length binary string. The VARBINARY function returns a varying-length binary string representation of a string of any type.

Related concepts

"BINARY data type and function" on page 5

Enhancements to optimistic currency control and update detection

Enhancements to optimistic concurrency control provide a faster, more scalable locking alternative to database locking for concurrent data access. An additional related enhancement provides a mechanism to detect recent (daily, weekly, or monthly) database updates.

Optimistic locking

Optimistic locking minimizes the time for which a given resource is unavailable for use by other transactions.

Because DB2 can determine when a row was changed, it can ensure data integrity while limiting the time that locks are held. With optimistic concurrency control, DB2 releases the row or page locks immediately after a read operation. To ensure data integrity, DB2 also releases the row lock after each FETCH operation and takes a new lock on a row only for a positioned update or delete.

A new row change timestamp column that you specify on the CREATE TABLE and ALTER TABLE statement lets you implement optimistic concurrency control. The column is defined with one of two options:

- NOT NULL GENERATED ALWAYS FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP
- NOT NULL GENERATED BY DEFAULT FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP.

DB2 maintains the contents of the row change timestamp column. When you want to use this change token as a condition when making an update, you can specify an appropriate condition for this column in your WHERE clause.

Daily, weekly, and monthly update detection

Database administrators benefit from knowing the volume of updates within specific time ranges so that they can plan for data replication, create auditing scenarios, and so on.

A new expression, ROW CHANGE, returns a token or a timestamp that represents the last change to a row. Now, an application has the following options:

- Determine when a row was last changed (or changed within a range of dates or number of days) by using the ROW CHANGE TIMESTAMP expression
- Return a token as a BIGINT (big integer) value that represents a relative point in the modification sequence of a row by using the ROW CHANGE TOKEN expression

MERGE and SELECT FROM MERGE statements

In prior releases, merging data (from 100 transactions to a master table, for example) required many separate operations:

- 100 update operations to update existing rows in the master table
- Insert operation to insert zero to 100 rows from the transactions that do not currently exist in the master table

V9.1 provides an easier and more efficient mechanism through the MERGE statement using arrays of input. The MERGE statement lets you update and insert many rows in a table from a single statement. You can embed the MERGE statement in an application program or issue it interactively. The statement is executable and can be dynamically prepared. In addition, you can use the SELECT

FROM MERGE statement to return all the updated rows and inserted rows, including column values that are generated by DB2.

Tools that support your enterprise

IBM offers many tools that help you perform administration tasks, access and integrate your information, and help you more effectively manage utilities, performance, recovery, and applications.

Accessing your enterprise data on demand with DB2 QMF

DB2 QMF Version 9.1 offers a completely redesigned, cross-platform workstation and Web-based solution that provides on demand access to data, reports, and interactive visual solutions through a rich desktop application or an ordinary Web browser. In addition, this latest release of DB2 QMF supports the new function in DB2 V9.1 for z/OS.

The DB2 QMF family of products includes:
- DB2 QMF for TSO/CICS
- DB2 QMF High Performance Option (HPO)
- DB2 QMF for Workstation
- DB2 QMF for WebSphere

DB2 QMF Version 9.1 provides the following enhancements:
- Support for DB2 Version 9.1 for z/OS, including XML and enhancements to SQL
- Drag-and-drop development of OLAP analytics, SQL queries, tabular reports, graphical reports, pivot tables, and data analysis views
- Drag-and-drop development of interactive visual solutions such as executive dashboards that are capable of graphically rendering data that is simultaneously mined from multiple data sources
- A rich security model that supports the personalization and dissemination of data and reports on a per-user or per-group basis
- A new object repository that simplifies the storage, access control, and distribution of QMF objects across the enterprise
- A re-engineered Eclipse-based desktop application that extends QMF to numerous workstation operating systems, including Windows, Linux, Solaris, and AIX®
- Full assimilation of the former QMF Visionary solution into the core DB2 QMF product, which extends the Visionary graphical objects to QMF reports and unifies the development of visual solutions within a single application
- A redesign of DB2 QMF for WebSphere, which extends key QMF functionality to browser-based users across multiple platforms

DB2 QMF for TSO/CICS

DB2 QMF for TSO/CICS Version 9.1 offers significant enhancements in database processing, support for new DB2 V9.1 for z/OS data types, import and export, security, and new diagnostic capabilities.

Some of the highlights of Version 9.1 of DB2 QMF for TSO/CICS include:
- Enhancements in database processing include a new program parameter that takes advantage of DB2 multiple-row fetch and insert processing.
- Support for the following new DB2 V9.1 for z/OS data types:
  - BIGINT

  - BINARY
  - DECFLOAT
  - VARBINARY
  - XML
- Enhancements to import and export functions, including:
  - Support for new XML data format
  - Ability to export and import data using UNIX path names
  - Ability to export and import QMF objects and data using PDSE data sets
  - Easier specification of data set size
- Enhancements to security through the QMF CONNECT command, which now supports RACF mixed-case passwords
- Diagnostic support for several new DB2 V9.1 for z/OS SQL codes

DB2 QMF for Workstation

The former DB2 QMF for Windows has been re-engineered and renamed DB2 QMF for Workstation.

The new Eclipse-based architecture enables DB2 QMF for Workstation to support popular workstation environments such as Linux, UNIX, Solaris, and AIX, in addition to Windows.

Other changes and enhancements to DB2 QMF for Workstation include:

- Full integration of the former DB2 QMF Visionary product into DB2 QMF for Workstation
- A new object repository that replaces the former Server Definitions File (SDF) and offers a variety of enhancements
- Several productivity and ease-of-use enhancements, including:
  - Enhanced support for OLAP
  - Re-engineered table editing capabilities
- A distributed architecture that:
  - Enables users to access data sources remotely by using network or Web-service connections over HTTP/HTTPS
  - Enables administrators to optionally consolidate database access on a designated server or collection of servers
  - Blends the benefits of a zero client-side administration while retaining the flexibility and performance of a rich desktop application

DB2 QMF for WebSphere

DB2 QMF for WebSphere provides dramatic improvements in performance and usability. Key enhancements include:

- A redesigned user interface with improved performance and reduced server response time, and a layout that is similar to DB2 QMF for Workstation
- Full support for capabilities that were formerly provided in DB2 QMF Visionary
- A fully independent Web-based QMF administration console that allows QMF administrators to manage the QMF for WebSphere application through an ordinary Web browser
- Support for new and existing data types and new SQL extensions in DB2 V9.1 for z/OS, including:
  - BIGINT
  - BINARY

- DECFLOAT
- VARBINARY
- XML

DB2 QMF High Performance Option

Key enhancements for Version 9.1 of the DB2 QMF High Performance Option (HPO) include:

- Leverage of DB2 multi-row fetch and insert for increased performance
- Support for DSQLLIB to allocate libraries, providing support that is comparable to that of QMF for TSO
- A new option to automatically replace an existing stored procedure
- Incorporates a stored procedure interface that takes a query and form as input, and returns the resulting report as a result set
- Adds support for the new BIGINT DB2 data type

Packaging options for DB2 QMF

Two packaging options for DB2 QMF are available: DB2 QMF Enterprise Edition and DB2 QMF Classic Edition.

The DB2 QMF Enterprise Edition provides the entire DB2 QMF family of technologies, which enables enterprise-wide business information across end-user and database platforms. DB2 QMF Enterprise Edition consists of the following components:

- DB2 QMF for TSO/CICS
- DB2 QMF High Performance Option (HPO)
- DB2 QMF for Workstation
- DB2 QMF for WebSphere

The DB2 QMF Classic Edition supports end users that access DB2 for z/OS databases solely from traditional mainframe terminals and emulators, including IBM Host On Demand. This edition consists of DB2 QMF for TSO/CICS.

Managing your enterprise with DB2 Tools

DB2 Tools for z/OS help reduce manual tasks, maintain high availability, and perform information replication and integration functions.

Together, the array of tools that IBM offers can help you meet performance standards and control IT costs. You can combine a selection of tools from any category to build a complete, flexible, and affordable solution that is tailored to meet your specific needs. A wide variety of tools are available and ready to support all major new V9.1 functions.

Database administration

Database administration tools meet common database service and support requirements to streamline the management of DB2. DB2 Database Administration tools include:

- IBM DB2 Administration Tool for z/OS
- IBM DB2 Administration Toolkit for z/OS the SAP Edition
- IBM DB2 Audit Management Expert for z/OS
- IBM DB2 Object Comparison Tool for z/OS
- IBM DB2 Storage Management Utility for z/OS

- IBM DB2 Thread Expert for z/OS
- IBM Data Encryption for IMS and DB2 Databases for z/OS

Utilities management

Utilities management tools provide a full set of utilities that handle unload, load, and many other tasks. Numerous enhanced utilities help improve productivity, performance, and availability. Utilities management tools include:
- IBM DB2 Automation Tool for z/OS
- IBM DB2 Automation Toolkit for z/OS the SAP Edition
- IBM DB2 Cloning Tool for z/OS
- IBM DB2 High Performance Unload for z/OS
- IBM DB2 Utilities Suite for z/OS

Important: In this version of DB2 for z/OS the DB2 Utilities Suite is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them. See Part 1 of *DB2 Utility Guide and Reference* for packaging details.

Performance management

Performance management tools help you keep DB2 running at peak levels even under heavy demand. Performance management tools include:
- IBM DB2 Buffer Pool Analyzer for z/OS
- IBM DB2 Optimization Expert for z/OS
- IBM DB2 Performance Toolkit for DB2 on z/OS the SAP Edition
- IBM DB2 Query Monitor for z/OS
- IBM DB2 SQL Performance Analyzer for z/OS
- IBM Tivoli® OMEGAMON® XE for DB2 Performance Expert on z/OS
- IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS

Recovery management

Recovery management tools manage a variety of image copy and backup and recovery tasks, including change accumulation, that improve the flexibility of DB2 and increase the value and availability of copied data for DB2. Recovery management tools include:
- IBM DB2 Archive Log Accelerator for z/OS
- IBM DB2 Change Accumulation Tool for z/OS
- IBM DB2 Log Analysis Tool for z/OS
- IBM DB2 Object Restore for z/OS
- IBM DB2 Recovery Expert for z/OS
- IBM Application Recovery Tool for IMS and DB2 Databases

Application management

Application management tools help manage DB2 applications and offer cost-effective global access to information and Web-enabled transactions. Application management tools include:
- IBM DB2 Bind Manager for z/OS
- IBM DB2 Data Archive Expert for z/OS
- IBM DB2 Path Checker for z/OS
- IBM DB2 Table Editor for z/OS
- IBM DB2 Test Database Generator for z/OS

Business analysis

Business analysis tools help business analysts leverage new business intelligence from their database data. Business analysis tools include:

- IBM DataQuant for z/OS
- IBM QMF for z/OS
- IBM DB2 Web Query Tool for z/OS

Information integration

Information integration tools offer enhanced data replication and data event publishing capabilities as part of the full Information Integration Solutions portfolio. Information integration tools include:

- IBM WebSphere Classic Data Event Publisher for z/OS
- IBM WebSphere Classic Federation Server for z/OS
- IBM WebSphere Classic Replication Server for z/OS
- IBM WebSphere Data Event Publisher for z/OS
- IBM WebSphere Data Integration Classic Connector for z/OS
- IBM WebSphere DataStage® for z/OS
- IBM WebSphere QualityStage for z/OS
- IBM WebSphere Replication Server for z/OS

Related information

DB2 and IMS Tools Web page
For more information about DB2 tools, see the IBM DB2 and IMS Tools Web page at http://www.ibm.com/software/data/db2imstools/

Chapter 3. Availability

V9.1 offers improved availability through a new DB2 command and extensions to existing SQL statements and online utilities.

Online REORG with no BUILD2 phase

When you reorganize a table space with SHRLEVEL CHANGE in previous versions of DB2, the data that is undergoing reorganization is not available to applications during the relatively lengthy SWITCH and BUILD2 phases. Now, in V9.1, the BUILD2 phase has been eliminated to improve availability of your data.

Faster replacement of one table with another

V9.1 provides the ability to replace tables faster through support for clone tables.

With clone table support, you can generate a copy of a current table, in the same table space, that has the same attributes, structure, and data as the original table. After you create a clone table, you can insert or load data into the clone table and exchange the clone table name with the current table name.

Extensions to the ALTER TABLE statement support the clone table function and let you:

- Create clone tables with the ADD CLONE option. Clone tables can be partitioned or non-partitioned, and you can make image copies of a clone table.
- Drop clone tables with the DROP CLONE option.
- Exchange current data with clone data with the EXCHANGE option, which provides a fast replacement of the original data, and is an alternative to the online LOAD REPLACE capability.

Web-based applications that need maximum availability can benefit from the option to implement a pair of tables that have cloned structures. Copies for application testing and auditing can be easily created. Clone table support also provides the unique ability to quickly change table names. As a result, applications can quickly and almost transparently switch between dual mirror tables. When the data in the clone table needs to become active to an application, you can use the EXCHANGE option of the ALTER TABLE statement to switch the table name, which provides fast replacement of the original data.

Universal table spaces

V9.1 introduces a new type of table space—a *universal table space*. A universal table space is a table space that is both segmented and partitioned. Two types of universal table spaces are available: the partition-by-growth table space and the range-partitioned table space.

A universal table space offers the following benefits:

- Better space management relative to varying-length rows: A segmented space map page provides more information about free space than a regular partitioned space map page.

© Copyright IBM Corp. 2007

- Improved mass delete performance: Mass delete in a segmented table space organization tends to be faster than table spaces that are organized differently. In addition, you can immediately reuse all or most of the segments of a table.

Partition-by-growth table space

Before V9.1, partitioned tables required key ranges to determine the target partition for row placement. Partitioned tables provide more granular locking and parallel operations by spreading the data over more data sets. Now, in V9.1, you have the option to partition according to data growth, which enables segmented tables to be partitioned as they grow, without the need for key ranges. As a result, segmented tables benefit from increased table space limits and SQL and utility parallelism that were formerly available only to partitioned tables, and you can avoid needing to reorganize a table space to change the limit keys.

You can implement partition-by-growth table space organization in several ways:

- You can use the new MAXPARTITIONS clause on the CREATE TABLESPACE statement to specify the maximum number of partitions that the partition-by-growth table space can accommodate. The value that you specify in the MAXPARTITIONS clause is used to protect against run-away applications that perform an insert in an infinite loop.
- You can use the MAXPARTITIONS clause on the ALTER TABLESPACE statement to alter the maximum number of partitions to which an existing partition-by-growth table space can grow. This ALTER TABLESPACE operation acts as an immediate ALTER.

Range-partitioned table space

A *range-partitioned table space* is a type of universal table space that is based on partitioning ranges and that contains a single table.

The new range-partitioned table space does not replace the existing partitioned table space, and operations that are supported on a regular partitioned or segmented table space are supported on a range-partitioned table space. You can create a range-partitioned table space by specifying both SEGSIZE and NUMPARTS keywords on the CREATE TABLESPACE statement.

With a range-partitioned table space, you can also control the partition size, choose from a wide array of indexing options, and take advantage of partition-level operations and parallelism capabilities. Because the range-partitioned table space is also a segmented table space, you can run table scans at the segment level. As a result, you can immediately reuse all or most of the segments of a table after the table is dropped or a mass delete has been performed.

Better availability during REBUILD INDEX operations

The online REBUILD INDEX utility has been extended in V9.1 to support read and write access for a longer period of time during the utility operation. As a result, applications have greater access to data while indexes on that data are being rebuilt. This complements the Version 8 support in which insert, update, and delete operations are supported on indexes that are non-unique while the index is in the process of rebuilding.

Improved availability with column and index renaming capabilities

Two new capabilities allow you to rename a column without needing to drop and rebuild a table, and to rename an index through a catalog operation.

The ALTER TABLE statement has been expanded to include the RENAME COLUMN clause. Now, you can rename a column without needing to drop and rebuild a table, which is less time consuming, reduces processing costs, and provides a higher degree of availability. The ability to rename a column in this manner also allows for better reuse of columns because you do not need to add additional columns.

The RENAME statement has been expanded to include the INDEX keyword, which lets you rename an index through a catalog operation. The ability to rename indexes in this manner also lets you consolidate index names that are used in different releases of SAP.

Modify EARLY code without an IPL

The new DB2 command, REFRESH DB2,EARLY, lets you refresh EARLY code when DB2 is not active. No IPL is required to complete the refresh operation. When you enter the command, the EARLY code modules are reloaded, and the EARLY control block is rebuilt. Previous copies of the EARLY modules are deleted the next time that you start DB2.

ALTER TABLESPACE and index logging improvements

The ability to alter a table space to suppress logging is extended in V9.1 to base table spaces, XML table spaces, and the indexes that are associated with them. Indexes inherit logging attributes from the base table with which they are associated. Suppression of logging is advantageous in a number of situations in which data is being duplicated. In those cases, if the data is lost, you can regenerate it from the original source rather than from an image copy and subsequent application of log records.

Related concepts

"Expanded support for not logging table spaces" on page 7

Support for using SMS storage classes with DB2-defined data sets

Users of DB2 for z/OS have asked for the ability to use SMS storage classes with DB2-defined data sets through the DB2 STOGROUP statement. V9.1 now supports this ability. The CREATE STOGROUP and ALTER STOGROUP statements have been enhanced to include SMS data class, management class, and storage class as optional parameters. Enabling DB2-defined data sets to use SMS storage classes increases the flexibility of handling these data sets while minimizing the time-consuming manual effort that is involved.

Chapter 4. Performance

V9.1 provides improved performance through reductions in CPU processing time for a number of utilities, faster disk access, improved logging and insert operations, improved query optimization, and index improvements.

Reduction in CPU processing time for utilities

CPU processing time is now reduced when you extract non-padded index keys from data rows by using the following utilities:

- COPY
- REORG
- LOAD
- REBUILD INDEX
- RECOVER
- RUNSTATS
- CHECK INDEX

SQL optimization improvements

V9.1 offers a number of SQL optimization enhancements, including support for histogram statistics, improved optimization techniques, and better management with the new DB2 Optimization Services Center.

Better data for the optimizer with histogram statistics

Data distribution statistics are important for query optimization. DB2 chooses the best access path based on cost. The basic foundation of cost is predicate selectivity estimation, which relies heavily on data distribution statistics. Prior versions of DB2 for z/OS rely on frequency statistics that are collected (through the RUNSTATS utility) on single values, either from a single column or from multiple columns. Now, V9.1 supports histogram statistics, which provide better data for the optimizer. With histogram statistics, DB2 can improve access path selection by estimating predicate selectivity from value-distribution statistics that are collected over the entire range of values in a data set, unlike frequency statistics.

Improved optimization techniques

DB2 V91. for z/OS adds or improves several techniques for optimizing query performance.

Global query optimization

In prior versions of DB2 for z/OS, query performance problems can result when DB2 breaks a query into multiple parts (for example, a correlated subquery and an outer query) and optimizes each of those parts independently. While each of the individual parts might be optimized to run efficiently, the overall result can be inefficient when these parts are combined. V9.1 now supports global query optimization, which enables DB2 to optimize a query as a whole rather than as an independent part. When a query is optimized as a whole entity, DB2 can consider the effect of one query block on another, and can consider reordering query blocks to determine the most efficient query path.

© Copyright IBM Corp. 2007

Sparse index and in-memory data caching enhancements

In Version 8 of DB2 for z/OS, materialized work files for star join queries can be stored in the global storage pool above the 2-GB bar (an in-memory data cache) as more cost-effective alternative to the sparse index technique that is used in prior versions of DB2 for z/OS. A sparse index is used as a fallback at runtime if enough memory is not available in the global storage pool.

In V9.1, the use of in-memory data caching is generalized to support more types of queries. In addition, data is cached in a local storage pool above the 2-GB bar, which can reduce potential storage contention because data caching storage management is associated with each thread. You can allocate up to 20 MB of memory for data caching. If the required memory exceeds the specified limit, a sparse index is used instead. This expanded access method provides potential benefits for tables that lack an appropriate index or enough statistics, and competes with other options for the most efficient access path.

Complex query optimization through page-range screening enhancements

The number of partitions that DB2 accesses to evaluate a query predicate can affect the performance of that query. A query that provides data retrieval through a data-partitioned secondary index (DPSI) might access some or all partitions of the DPSI. For a query that is based only on a DPSI key value or range, DB2 must examine all partitions. However, if the query also has predicates on the leading columns of the partitioning key, DB2 does not need to examine all of the partitions. Version 8 of DB2 for z/OS introduced page-range screening or, limited partition scan, as a technique for removing inapplicable partitions from consideration for access. Several techniques for writing or modifying queries to take advantage of page-range screen were also offered.

V9.1 offers two additional enhancements to the page-range screening function that further reduce the number of partitions that are accessed unnecessarily. Now you can take advantage of page-range screening by writing queries that join the tables on the partitioning column or create a non-matching predicate on the partitioning keys.

Autonomic reoptimization for dynamic queries

The access path for static and dynamic queries that contain host variables is chosen by the optimizer during bind time, before the values of the host variables are available. Because the values of the host variables are not considered, the access path that is chosen might not always be optimal. In versions prior to Version 8 of DB2 for z/OS, this problem is partially solved by the REOPT(ALWAYS) bind option, which prepares the statement again at run time when the host input variables are available. As a result, the optimizer can reoptimize the access path based on known input variable values. However, frequently called SQL statements that take little time to execute are not good candidates for this method.

Version 8 of DB2 for z/OS introduced the bind option, REOPT(ONCE), which reoptimizes the access path of a dynamic query only once at run time, no matter how many times the same statement is executed. With this technique, DB2 chooses the access path based on the set of input variable values, and the access path is stored in the dynamic statement cache for use by subsequent executions of the query. While this option offers additional flexibility, the impact of host variables on the selection of optimal access paths continues to be observed.

V9.1 now offers greater flexibility with the REOPT(AUTO) option, which enables DB2 to autonomically determine if a new access path is needed for a statement in the dynamic statement cache. When you specify REOPT(AUTO), DB2 optimizes the access path for dynamic SQL statements at the first execute or open operation. Each time a statement is executed, DB2 determines whether a new access path is needed to improve the performance of the statement. If a new access path will improve the performance, DB2 generates a new access path and replaces the previous access path in the dynamic statement cache.

FETCH FIRST *n* ROWS ONLY and ORDER BY in subselects and fullselects

In prior versions of DB2 for z/OS, the ORDER BY and FETCH FIRST *n* ROWS ONLY clauses were supported only at the statement level as part of select-statement or a SELECT INTO statement. V9.1 delivers additional flexibility by allowing both FETCH FIRST *n* ROWS ONLY and ORDER BY clauses when they are specified as part of a subselect or a fullselect statement. With the new support for FETCH FIRST *n* ROWS ONLY, you can select the top *n* rows in the result table of a table expression, a leg of a union, or a subquery (for a fullselect that is a component of a predicate). If you specify the FETCH FIRST *n* ROWS ONLY clause with the ORDER BY clause, the ordering is performed on the entire result table before the first *n* rows are returned.

IBM Optimization Service Center for DB2 for z/OS

IBM Optimization Service Center for DB2 for z/OS (Optimization Service Center) is a workstation tool that provides an innovative set of capabilities to improve the performance of individual DB2 SQL queries and entire SQL workloads that run on your DB2 for z/OS subsystem. You can use Optimization Service Center to identify and analyze problem SQL statements and receive expert advice about statistics that you might gather to improve the performance of an individual statement.

For individual queries, Optimization Service Center can:
- Annotate the query
- Draw an access plan graph
- Generate query reports
- Facilitate generation of optimization hints
- Suggest statistics needed by the SQL optimizer

At your request, Optimization Service Center can send query-related information back to IBM service.

For SQL workloads, Optimization Service Center can gather workloads from various sources such as the dynamic statement cache, catalog tables, files and directories, QMF tables, and profile monitoring.

Additionally, Optimization Service Center can:
- Capture SQL queries immediately or at a scheduled time, either periodically or on a one-time basis.
- Suggest important statistics to be collected to improve performance for the entire workload.
- Manage and use profile monitoring. Profile monitoring provides a lightweight exception monitoring mechanism in which only exception statements are pushed out by DB2.

OSC is offered as part of the DB2 Accessories Suite for z/OS, an orderable, no-charge feature of DB2 Version 9.1 for z/OS.

IBM DB2 Optimization Expert for z/OS

IBM DB2 Optimization Expert for z/OS (DB2 Optimization Expert) is a workstation tool which offers a comprehensive set of index and statistics advisors that you can use to improve system performance and lower your total cost of ownership. In addition to a powerful set of expert advisor tools, DB2 Optimization Expert includes all of the Optimization Service Center functions.

With DB2 Optimization Expert, you can:

- View the query activity to find performance problems from a snapshot of the statement cache of your DB2 subsystem and a number of other sources (the DB2 catalog, for example)
- Get and implement expert tuning recommendations to improve the performance of an SQL query by using some or all of the following expert advisors:
  - Statistics advisor, which recommends statistics to update or collect to improve the performance of a query
  - Query advisor, which recommends ways to rewrite an SQL query to improve performance
  - Access path advisor, which alerts you to problematic access paths in the access plan for the query that might cause poor performance
  - Index advisor, which recommends new indexes to enhance the performance of an SQL query and generates CREATE INDEX statements that you can run from your workstation to implement the recommendations.
  - Optimization Service Center functions, which support advanced tuning with powerful optimization tools.

    All the Optimization Service Center functions of the DB2 Accessories Suite are included with DB2 Optimization Expert. In summary, they enable experienced DBAs to understand, analyze, format, and optimize SQL statements.
- View the activity of groups of SQL statements to find performance problems: You can monitor the health of the SQL workloads. When the performance of SQL queries have been optimized, you can create monitor profiles for normal processing and exception processing that:
  - Monitor the health of SQL processing on the subsystem
  - Alert you when problems develop
  - Alert you when more tuning activities might be advised
- Get expert tuning advice from the following expert advisor functions to improve the performance of an entire SQL workload:
  - Workload statistics advisor, which recommends statistics to collect or update to improve the overall performance of the statements that make up an SQL workload.
  - Workload index advisor, which recommends indexes to create or modify to improve the overall performance of the statements that make up an SQL workload. The workload index advisor also generates CREATE INDEX statements that you can run from your workstation to implement the recommendations.
  - Workload query advisor, recommends ways to rewrite specific SQL queries to improve performance of a workload. Best-practice fixes to common query-writing mistakes are recommended.

DB2 Optimization Expert for z/OS is a separately orderable, priced DB2 Tool.

Indexing improvements

Indexing improvements contribute to the overall improvements in query performance. Specific improvements include index compression, index on expression, index key randomization, and larger index page sizes.

Index compression

V9.1 provides the ability to compress an index without using a dictionary. Compressing an index reduces the physical storage space that an index requires. By eliminating the need for a dictionary, index data can be compressed as soon as the first index entries are added to an index, and the space that would normally be used by a dictionary becomes available. You can choose whether you want to use index compression by specifying COMPRESS YES or COMPRESS NO on the CREATE INDEX or the ALTER INDEX statements.

Index on expression

Support for a new index type, index on expression, lets you create an index on a general expression. Query performance can be enhanced if the optimizer chooses that index. When you use an index on an expression, the results of the expressions are evaluated during insertion time or during an index rebuild and are kept in the index. If the optimizer chooses to use that index, the predicate is evaluated against the values that are stored in the index. As a result, run-time performance overhead is eliminated.

Larger index page sizes

Prior versions of DB2 for z/OS limit the size of an index page to 4 KB. The size of an index page limits the number of index keys that the index page can accommodate and can cause contention in indexes that split frequently. V9.1 lifts these restrictions by offering expanded index page sizes of 8 KB, 16 KB, and 32 KB. An index page size that is greater than 4 KB accommodates more index keys per page and can reduce the frequency of index page splits. You can use the INDEXBP option on both CREATE DATABASE and ALTER DATABASE statements to specify 4-KB, 8-KB, 16-KB, or 32-KB index buffer pools. You can also use the BUFFERPOOL keyword on the CREATE INDEX statement to specify 8-KB, 16-KB, and 32-KB buffer pools.

Index key randomization

Currently, online transaction processing (OLTP) workloads in a data sharing environment can experience locking contention on the last page of an index, especially when an application program uses indexes on columns that contain current timestamps or ever-increasing sequential values create insertion hot spots. As a result of this contention, an application must wait to acquire an index page. To offer relief from locking contention, V9.1 provides support to randomize index key columns through the new RANDOM option on both CREATE INDEX and ALTER INDEX statements.

When you specify the RANDOM option, values are stored at random places in the index tree. As a result, the number of consecutive insertions on a page decreases and contention is also decreased. You can use a randomly ordered index to perform equality lookups on a specified column. In addition, key columns that are

in random order can be used in non-matching index scans, and index-only access on random key columns is possible. Even though values are stored in random order, you can retrieve the original value of the random key column.

Related concepts

"Capability to create an index on an expression" on page 9

"Relief for sequential key insert"

Improved performance for varying-length rows

Prior to V9.1, if you store a value that is shorter than the length of a column in a varying-length column, the data is not padded to the full length of the column. As a result, columns that follow varying-length columns are at variable offsets in the row. When you need to locate and access such a column, you must scan the columns sequentially after the first varying-length column. In V9.1, the format in which a row that contains varying-length columns is stored in the table has been changed to facilitate locating columns within the row for data retrieval and predicate evaluation. As a result, you no longer need to run a sequential scan, and the performance improves for access to data in tables that store rows with varying-length columns.

Relief for sequential key insert

In prior releases, an index page splits so that approximately half of the index keys on the splitting page remain on one page, and the remaining index keys move to a new page. This 50:50 split can cause frequent page splits on an index that has sequential insert patterns. As a result, half of the splitting pages are empty. Non-partitioned indexes that are updated by LOAD utility jobs that run in parallel against multiple partitions are especially susceptible to these problems because sequential inserts into multiple ranges in the non-partitioned index can occur. Two enhancements in V9.1 alleviate these problems: support for asymmetric splitting of index pages and increased index page sizes (larger than 4 KB).

Allowing index pages to split asymmetrically can improve space utilization and reduce contention that results from frequent page splits in an index with sequential insert patterns in the middle of the index. An index page size that is greater than 4 KB can also relieve contention by accommodating more index keys per page, which reduces the frequency of page splits in indexes. You can use the INDEXBP option on both CREATE DATABASE and ALTER DATABASE statements to specify 4-KB, 8-KB, 16-KB, or 32-KB index buffer pools, and the BUFFERPOOL keyword on the CREATE INDEX statement to specify 8-KB, 16-KB, and 32-KB buffer pools.

Improved logging performance

Several logging enhancements in V9.1 improve performance:

- In prior versions, data sharing workloads that perform a high volume of logging can experience significant delays because of the extra processing required to set unique LRSN (log record sequence number) values for each record. In V9.1, the process for setting and updating LRSN values has been streamlined to improve performance.
- Version 8 of DB2 for z/OS and prior releases use the basic direct access method (BDAM) to read archive logs from disk, which facilitates rapid access to the VSAM control intervals, but which does not support striped data sets and extended format data sets. V9.1 replaces BDAM with the basic sequential access

method (BSAM), which supports both striped and extended format data sets, and improves read and write performance.

- In prior versions of DB2 for z/OS, an active log can have up to 4 GB of tracks on a single disk volume, whereas the size limit for an archive log is only about 3 GB. As a result, installations that use 4-GB active logs are forced to put archive logs on tape. Version 1.7 of z/OS now supports large data sets and removes the 64-KB archive log limit. V9.1 takes advantage of the new limit by introducing the DSNTYPE=LARGE attribute of the PRIMARY QUANTITY field of installation panel DSNTIPA. The new attribute supports an archive log that has up to 4 GB of tracks per disk volume.

Improved data insert performance

The APPEND option of the CREATE TABLE and ALTER TABLE statement offers increased performance for inserting data into the end of a table. This option reduces the processing that is used to target the locations for new rows

Like the LOAD RESUME utility, the rows are placed at the end of the table and are not clustered as they would be for a normal insert operation. The APPEND option provides a tradeoff that favors inserts rather than retrievals and the need to reorganize.

Other V9.1 functions that can improve data insert performance include several index improvements, logging performances improvements, and not logging table spaces.

Related concepts

"ALTER TABLESPACE and index logging improvements" on page 23

"Indexing improvements" on page 29

"Improved logging performance" on page 30

Chapter 5. Regulatory compliance

Regulatory compliance, security and auditing are in the headlines and growing much more important. DB2 9 helps you respond to increasing needs to protect security, to assure integrity, and to comply with regulations such as the Sarbanes-Oxley Act, the Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry (PCI) Security Standards. Improved access control with network trusted context and roles allows more precise control of security. Improved filtering makes auditing more usable. Secure Sockets Layer (SSL) data encryption on networks is more secure.

Roles and network trusted contexts

Support for trusted context addresses the problem of establishing a trusted relationship between DB2 and an external entity such as a database administrator or a middleware server. With trusted context support, a series of trusted attributes are evaluated to determine whether a specific context can be trusted. After a trusted context is established, you can define a unique set of interactions between DB2 and the external entity, such as a middleware server, so that the existing database connection can be used by a different user without requiring authentication of the new connection user.

Support for trusted context also provides the ability, within a specific trusted context, for a DB2 authorization ID to acquire a special set of privileges that are not available outside that trusted context by defining roles. A role is a database entity that groups together one or more privileges and can be assigned to users. A role provides privileges, in addition to the current set of privileges, that are granted to the primary and secondary authorization identifiers. A role can own objects if the objects are created in a trusted context with the role defined as the owner. If a role is defined as an owner, then only the privileges that are granted to the role are considered for object ownership.

Improved auditing

Improved trace filtering makes the jobs of auditing and of performance management easier. Many more options can be used to minimize the amount of data collected, so that the overhead is reduced and the extraneous data does not need to be processed.

Support for Secure Socket Layer protocol

DB2 Version 9.1 for z/OS supports the Secure Socket Layer (SSL) protocol by implementing the z/OS Communications Server IP Application Transparent Transport Layer Security (AT-TLS) function.

The z/OS V1R7 Communications Server for TCP/IP introduces the AT-TLS function in the TCP/IP stack for applications that require secure TCP/IP connections. AT-TLS performs transport layer security on behalf of the application, such as DB2, by invoking the z/OS system SSL in the TCP layer of the TCP/IP stack. The z/OS system SSL provides support for TLS V1.0, SSL V3.0, and SSL V2.0 protocols.

© Copyright IBM Corp. 2007

More security options with INSTEAD OF triggers

INSTEAD OF triggers, which are defined on views, provide another way of ensuring security within the database. INSTEAD OF triggers are used to process insert, update, and delete operations (through trigger logic) instead of the INSERT, UPDATE, or DELETE statement that activates the trigger. In addition to other capabilities, you can use INSTEAD OF triggers to encode and decode data from the database within a view. The view might encapsulate decryption functions, while the INSTEAD OF triggers use the encryption functions to ensure security within the database.

Related concepts

"INSTEAD OF triggers" on page 5

Appendix. How to obtain DB2 information

This section provides information that you can use to find valuable information about the DB2 product:

- "DB2 on the Web"
- "DB2 publications"
- "DB2 education" on page 36
- "How to order the DB2 library" on page 36

DB2 on the Web

Stay current with the latest information about DB2. View the DB2 home page on the Web. News items keep you informed about the latest enhancements to the product. Product announcements, press releases, fact sheets, and technical articles help you plan your database management strategy.

You can view and search DB2 publications on the Web, or you can download and print many of the most current DB2 books. Follow links to other Web sites with more information about DB2 family and z/OS solutions. Access DB2 on the Web at the following Web site: www.ibm.com/software/db2zos.

DB2 publications

The publications for DB2 for z/OS are available in various formats and delivery methods. IBM provides mid-version updates in softcopy on the Web and on CD-ROM.

DB2 Information Center for z/OS solutions

DB2 for z/OS product information is viewable in the DB2 Information Center for z/OS solutions. The information center is a delivery vehicle for information about DB2 UDB for z/OS, IMS, QMF, and related tools. This information center enables users to search across related product information in multiple languages for data management solutions for the z/OS environment. Product technical information is provided in a format that offers more options and tools for accessing, integrating, and customizing information resources. The information center is based on Eclipse open source technology.

The DB2 Information Center for z/OS solutions is viewable at the following Web site: http://publib.boulder.ibm.com/infocenter/db2zhelp.

CD-ROMs and DVD

Books for DB2 V9.1 for z/OS are available on a CD-ROM that is included with your product shipment:

- DB2 V9.1 for z/OS Licensed Library Collection, LK3T-7195, in English

The CD-ROM contains the collection of books for DB2 V9.1 for z/OS in PDF and BookManager® formats. Periodically, IBM refreshes the books on subsequent editions of this CD-ROM.

The books for DB2 for z/OS are also available on the following CD-ROM and DVD collection kits, which contain online books for many IBM products:

© Copyright IBM Corp. 2007

- IBM z/OS Software Products Collection , SK3T-4270, in English
- IBM z/OS Software Products DVD Collection , SK3T–4271, in English

PDF format

Many of the DB2 books are available in PDF (Portable Document Format) for viewing or printing from CD-ROM or the Web. Download the PDF books to your intranet for distribution throughout your enterprise.

BookManager format

You can use online books on CD-ROM to read, search across books, print portions of the text, and make notes in these BookManager books. Using the IBM Softcopy Reader, appropriate IBM Library Readers, or the BookManager Read product, you can view these books in the z/OS, Windows, and VM environments. You can also view and search many of the DB2 BookManager books on the Web.

DB2 education

IBM Education and Training offers a wide variety of classroom courses to help you quickly and efficiently gain DB2 expertise. IBM schedules classes are in cities all over the world. You can find class information, by country, at the IBM Learning Services Web site: www.ibm.com/services/learning.

IBM also offers classes at your location, at a time that suits your needs. IBM can customize courses to meet your exact requirements. For more information, including the current local schedule, please contact your IBM representative.

How to order the DB2 library

You can order DB2 publications and CD-ROMs through your IBM representative or the IBM branch office that serves your locality. If your location is within the United States or Canada, you can place your order by calling one of the toll-free numbers:
- In the U.S., call 1-800-879-2755.
- In Canada, call 1-800-426-4968.

To order additional copies of licensed publications, specify the SOFTWARE option. To order additional publications or CD-ROMs, specify the PUBLICATIONS option. Be prepared to give your customer number, the product number, and either the feature codes or order numbers that you want.

You can also order books from the IBM Publication Center on the Web: www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.

From the IBM Publication Center, you can go to the Publication Notification System (PNS). PNS users receive electronic notifications of updated publications in their profiles. You have the option of ordering the updates by using the publications direct ordering application or any other IBM publication ordering channel. The PNS application does not send automatic shipments of publications. You will receive updated publications and a bill for them if you respond to the electronic notification.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

© Copyright IBM Corp. 2007

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

Company, product, or service names identified in the DB2 Version 9.1 for z/OS information may be trademarks or service marks of International Business Machines Corporation or other companies. Information about the trademarks of IBM Corporation in the United States, other countries, or both is located at http://www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies, and have been used at least once in the DB2 for z/OS library:

- Microsoft, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel®, Intel logo, Intel Inside®, Intel Inside logo, Intel Centrino™, Intel Centrino logo, Celeron®, Intel Xeon™, Intel SpeedStep®, Itanium®, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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 registered 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.

Glossary

basic row format
A row format in which values for columns are stored in the row in the order in which the columns are defined by the CREATE TABLE statement.

clone object
An object that is associated with a clone table, including the clone table itself and check constraints, indexes, and BEFORE triggers on the clone table.

clone table
A table that is structurally identical to a base table. The base and clone table each have separate underlying VSAM data sets, which are identified by their data set instance numbers.

DOCID
See document ID.

document ID
A value that uniquely identifies a row that contains an XML column. This value is stored with the row and never changes.

Extensible Markup Language (XML)
A standard metalanguage for defining markup languages that is a subset of Standardized General Markup Language (SGML).

file reference variable
A host variable that is declared with one of the derived data types (BLOB_FILE, CLOB_FILE, DBCLOB_FILE); file reference variables direct the reading of a LOB from a file or the writing of a LOB into a file.

histogram statistics
A way of summarizing data distribution. This technique divides up the range of possible values in a data set into intervals, such that each interval contains approximately the same percentage of the values. A set of statistics are collected for each interval.

IBM System z9 Integrated Processor (zIIP)
A specialized processor that can be used for some DB2 functions.

INSTEAD OF trigger
A trigger that is associated with a single view and is activated by an insert, update, or delete operation on the view and that can define how to propagate the insert, update, or delete operation on the view to the underlying tables of the view.

intersection
An SQL operation that involves the INTERSECT set operator, which combines two result tables. The result of an intersection operation consists of all of the rows that are in both result tables.

native SQL procedure
An SQL procedure that is processed by converting the procedural statements to a native representation that is stored in the database directory, as is done with other SQL statements. When a native SQL procedure is called, the native representation is loaded from the directory, and DB2 executes the procedure.

© Copyright IBM Corp. 2007

online analytical processing (OLAP)
The process of collecting data from one or many sources; transforming and analyzing the consolidated data quickly and interactively; and examining the results across different dimensions of the data by looking for patterns, trends, and exceptions within complex relationships of that data.

partition-by-growth table space
A table space whose size can grow to accommodate data growth. DB2 for z/OS manages partition-by-growth table spaces by automatically adding new data sets when the database needs more space to satisfy an insert operation.

partitioned table space
A table space that is based on a single table and that is subdivided into partitions, each of which can be processed independently by utilities.

partitioning index
An index in which the leftmost columns are the partitioning columns of the table. The index can be partitioned or nonpartitioned.

range-partitioned table space
A type of universal table space that is based on partitioning ranges and that contains a single table.

relational schema
See SQL schema.

reordered row format
A row format that facilitates improved performance in retrieval of rows that have varying-length columns. DB2 rearranges the column order, as defined in the CREATE TABLE statement, so that the fixed-length columns are stored at the beginning of the row and the varying-length columns are stored at the end of the row.

role
A database entity that groups together one or more privileges and that can be assigned to a primary authorization ID or to PUBLIC. The role is available only in a trusted context.

segmented table space
A table space that is divided into equal-sized groups of pages called segments. Segments are assigned to tables so that rows of different tables are never stored in the same segment.

SQL schema
A collection of database objects such as tables, views, indexes, functions, distinct types, schemas, or triggers that defines a database. An SQL schema provides a logical classification of database objects.

system authorization ID
The primary DB2 authorization ID that is used to establish a trusted connection. A system authorization ID is derived from the system user ID that is provided by an external entity, such as a middleware server.

trust attribute
An attribute on which to establish trust. A trusted relationship is established based on one or more trust attributes.

trusted connection
A database connection whose attributes match the attributes of a unique trusted context defined at the DB2 database server.

trusted connection reuse
The ability to switch the current user ID on a trusted connection to a different user ID.

trusted context
A database security object that enables the establishment of a trusted relationship between a DB2 database management system and an external entity.

trusted context default role
A role associated with a trusted context. The privileges granted to the trusted context default role can be acquired only when a trusted connection based on the trusted context is established or reused.

trusted context user
A user ID to which switching the current user ID on a trusted connection is permitted.

trusted context user-specific role
A role that is associated with a specific trusted context user. It overrides the trusted context default role if the current user ID on the trusted connection matches the ID of the specific trusted context user.

trusted relationship
A privileged relationship between two entities such as a middleware server and a database server. This relationship allows for a unique set of interactions between the two entities that would be impossible otherwise.

universal table space
A table space that is both segmented and partitioned.

WLM enclave
A construct that can span multiple dispatchable units (service request blocks and tasks) in multiple address spaces, allowing them to be reported on and managed by WLM as part of a single work request.

XML See Extensible Markup Language.

XML attribute
A name-value pair within a tagged XML element that modifies certain features of the element.

XML column
A column of a table that stores XML values and is defined using the data type XML. The XML values that are stored in XML columns are internal representations of well-formed XML documents.

XML data type
A data type for XML values.

XML element
A logical structure in an XML document that is delimited by a start and an end tag. Anything between the start tag and the end tag is the content of the element.

XML index
An index on an XML column that provides efficient access to nodes within an XML document by providing index keys that are based on XML patterns.

XML lock
A column-level lock for XML data. The operation of XML locks is similar to the operation of LOB locks.

XML node
The smallest unit of valid, complete structure in a document. For example, a node can represent an element, an attribute, or a text string.

XML node ID index
An implicitly created index, on an XML table that provides efficient access to XML documents and navigation among multiple XML data rows in the same document.

XML pattern
A slash-separated list of element names, an optional attribute name (at the end), or kind tests, that describe a path within an XML document in an XML column. The pattern is a restrictive form of path expressions, and it selects nodes that match the specifications. XML patterns are specified to create indexes on XML columns in a database.

XML publishing function
A function that returns an XML value from SQL values. An XML publishing function is also known as an XML constructor.

XML schema
In XML, a mechanism for describing and constraining the content of XML files by indicating which elements are allowed and in which combinations. XML schemas are an alternative to document type definitions (DTDs) and can be used to extend functionality in the areas of data typing, inheritance, and presentation.

XML schema repository (XSR)
A repository that allows the DB2 database system to store XML schemas. When registered with the XSR, these objects have a unique identifier and can be used to validate XML instance documents.

XML serialization function
A function that returns a serialized XML string from an XML value.

XML table
An auxiliary table that is implicitly created when an XML column is added to a base table. This table stores the XML data, and the column in the base table points to it.

XML table space
A table space that is implicitly created when an XML column is added to a base table. The table space stores the XML table. If the base table is partitioned, one partitioned table space exists for each XML column of data.

XSR See XML schema repository.

zIIP See IBM System z9 Integrated Processor.

Information resources for DB2 for z/OS and related products

Many information resources are available to help you use DB2 for z/OS and many related products. A large amount of technical information about IBM products is now available online in information centers or on library Web sites.

Disclaimer: Any Web addresses that are included here are accurate at the time this information is being published. However, Web addresses sometimes change. If you visit a Web address that is listed here but that is no longer valid, you can try to find the current Web address for the product information that you are looking for at either of the following sites:

- http://www.ibm.com/support/publications/us/library/index.shtml, which lists the IBM information centers that are available for various IBM products
- http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi, which is the IBM Publications Center, where you can download online PDF books or order printed books for various IBM products

DB2 for z/OS product information

The primary place to find and use information about DB2 for z/OS is the Information Management Software for z/OS Solutions Information Center (http://publib.boulder.ibm.com/infocenter/imzic), which also contains information about IMS, QMF, and many DB2 and IMS Tools products. The majority of the DB2 for z/OS information in this information center is also available in the books that are identified in the following table. You can access these books at the DB2 for z/OS library Web site (http://www.ibm.com/software/data/db2/zos/library.html) or at the IBM Publications Center (http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi).

Table 1. DB2 Version 9.1 for z/OS book titles

Title	Publication number	Available in information center	Available in PDF	Available in BookManager format	Available in printed book
DB2 Version 9.1 for z/OS Administration Guide	SC18-9840	X	X	X	X
DB2 Version 9.1 for z/OS Application Programming & SQL Guide	SC18-9841	X	X	X	X
DB2 Version 9.1 for z/OS Application Programming Guide and Reference for Java	SC18-9842	X	X	X	X
DB2 Version 9.1 for z/OS Codes	GC18-9843	X	X	X	X
DB2 Version 9.1 for z/OS Command Reference	SC18-9844	X	X	X	X
DB2 Version 9.1 for z/OS Data Sharing: Planning and Administration	SC18-9845	X	X	X	X
DB2 Version 9.1 for z/OS Diagnosis Guide and Reference [1]	LY37-3218		X	X	X

© Copyright IBM Corp. 2007

Table 1. DB2 Version 9.1 for z/OS book titles (continued)

Title	Publication number	Available in information center	Available in PDF	Available in BookManager format	Available in printed book
DB2 Version 9.1 for z/OS Diagnostic Quick Reference	LY37-3219				X
DB2 Version 9.1 for z/OS Installation Guide	GC18-9846	X	X	X	X
DB2 Version 9.1 for z/OS Introduction to DB2	SC18-9847	X	X	X	X
DB2 Version 9.1 for z/OS Licensed Program Specifications	GC18-9848		X		X
DB2 Version 9.1 for z/OS Messages	GC18-9849	X	X	X	X
DB2 Version 9.1 for z/OS ODBC Guide and Reference	SC18-9850	X	X	X	X
DB2 Version 9.1 for z/OS Performance Monitoring and Tuning Guide	SC18-9851	X	X	X	X
DB2 Version 9.1 for z/OS Optimization Service Center		X			
DB2 Version 9.1 for z/OS Program Directory	GI10-8737		X		X
DB2 Version 9.1 for z/OS RACF Access Control Module Guide	SC18-9852		X	X	
DB2 Version 9.1 for z/OS Reference for Remote DRDA Requesters and Servers	SC18-9853	X	X	X	
DB2 Version 9.1 for z/OS Reference Summary [2]	SX26-3854				
DB2 Version 9.1 for z/OS SQL Reference	SC18-9854	X	X	X	X
DB2 Version 9.1 for z/OS Utility Guide and Reference	SC18-9855	X	X	X	X
DB2 Version 9.1 for z/OS What's New?	GC18-9856	X	X	X	X
DB2 Version 9.1 for z/OS XML Extender Administration and Programming	SC18-9857	X	X	X	X
DB2 Version 9.1 for z/OS XML Guide	SC18-9858	X	X	X	X

Notes:

1. *DB2 Version 9.1 for z/OS Diagnosis Guide and Reference* is available in PDF and BookManager formats on the DB2 Version 9.1 for z/OS Licensed Collection kit, LK3T-7195. You can order this License Collection kit on the IBM Publications Center site (http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi). This book is also available in online format in DB2 data set DSN910.SDSNIVPD(DSNDR).
2. *DB2 Version 9.1 for z/OS Reference Summary* will be available in 2007.

Information resources for related products

In the following table, related product names are listed in alphabetic order, and the associated Web addresses of product information centers or library Web pages are indicated.

Table 2. Related product information resource locations

Related product	Information resources
C/C++ for z/OS	Library Web site: http://www.ibm.com/software/awdtools/czos/library/ This product is now called z/OS XL C/C++.
CICS Transaction Server for z/OS	Information center: http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp
COBOL	Information center: http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp This product is now called Enterprise COBOL for z/OS.
DB2 Connect™	Information center: http://publib.boulder.ibm.com/infocenter/db2luw/v9//index.jsp This resource is for DB2 Connect 9.
DB2 Database for Linux, UNIX, and Windows	Information center: http://publib.boulder.ibm.com/infocenter/db2luw/v9//index.jsp This resource is for DB2 9 for Linux, UNIX, and Windows.
DB2 Performance Expert for z/OS	Information center: http://publib.boulder.ibm.com/infocenter/imzic This product is now called DB2 Tivoli OMEGAMON for XE Performance Expert on z/OS.
DB2 Query Management Facility	Information center: http://publib.boulder.ibm.com/infocenter/imzic
DB2 Server for VSE & VM VSE	One of the following locations: • For VSE: http://www.ibm.com/support/docview.wss?rs=66&uid=swg27003758 • For VM: http://www.ibm.com/support/docview.wss?rs=66&uid=swg27003759
DB2 Tools	One of the following locations: • Information center: http://publib.boulder.ibm.com/infocenter/imzic • Library Web site: http://www.ibm.com/software/data/db2imstools/library.html These resources include information about the following products and others: • DB2 Administration Tool • DB2 Automation Tool • DB2 DataPropagator™ (also known as WebSphere Replication Server for z/OS) • DB2 Log Analysis Tool • DB2 Object Restore Tool • DB2 Query Management Facility • DB2 SQL Performance Analyzer • DB2 Tivoli OMEGAMON for XE Performance Expert on z/OS (includes Buffer Pool Analyzer and Performance Monitor)
DB2 Universal Database™ for iSeries™	Information center: http://www.ibm.com/systems/i/infocenter/
Debug Tool for z/OS	Information center: http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp
Enterprise COBOL for z/OS	Information center: http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp
Enterprise PL/I for z/OS	Information center: http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp
IMS	Information center: http://publib.boulder.ibm.com/infocenter/imzic

Table 2. Related product information resource locations (continued)

Related product	Information resources
IMS Tools	One of the following locations: • Information center: http://publib.boulder.ibm.com/infocenter/imzic • Library Web site: http://www.ibm.com/software/data/db2imstools/library.html These resources have information about the following products and others: • IMS Batch Terminal Simulator for z/OS • IMS Connect • IMS HALDB Conversion and Maintenance Aid • IMS High Performance Utility products • IMS DataPropagator • IMS Online Reorganization Facility • IMS Performance Analyzer
PL/I	Information center: http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp This product is now called Enterprise PL/I for z/OS.
System z	http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp
WebSphere Application Server	Information center: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp
WebSphere Message Broker with Rules and Formatter Extension	Information center: http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp The product is also known as WebSphere MQ Integrator Broker.
WebSphere MQ	Information center: http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp The resource includes information about MQSeries®.
WebSphere Replication Server for z/OS	Either of the following locations: • Information center: http://publib.boulder.ibm.com/infocenter/imzic • Library Web site: http://www.ibm.com/software/data/db2imstools/library.html This product is also known as DB2 DataPropagator.
z/Architecture™	Library Center site: http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

Table 2. Related product information resource locations (continued)

Related product	Information resources
z/OS	Library Center site: http://www.ibm.com/servers/eserver/zseries/zos/bkserv/ This resource includes information about the following z/OS elements and components: • Character Data Representation Architecture • Device Support Facilities • DFSORT™ • Fortran • High Level Assembler • NetView® • SMP/E for z/OS • SNA • TCP/IP • TotalStorage® Enterprise Storage Server® • VTAM® • z/OS C/C++ • z/OS Communications Server • z/OS DCE • z/OS DFSMS • z/OS DFSMS Access Method Services • z/OS DFSMSdss™ • z/OS DFSMShsm™ • z/OS DFSMSdfp™ • z/OS ICSF • z/OS ISPF • z/OS JES3 • z/OS Language Environment® • z/OS Managed System Infrastructure • z/OS MVS • z/OS MVS JCL • z/OS Parallel Sysplex® • z/OS RMF™ • z/OS Security Server • z/OS UNIX System Services
z/OS XL C/C++	http://www.ibm.com/software/awdtools/czos/library/

The following information resources from IBM are not necessarily specific to a single product:

- The DB2 for z/OS Information Roadmap; available at: http://www.ibm.com/software/data/db2/zos/roadmap.html
- DB2 Redbooks™ and Redbooks about related products; available at: http://www.ibm.com/redbooks
- IBM Educational resources:
  - Information about IBM educational offerings is available on the Web at: http://www.ibm.com/software/sw-training/

  - A collection of glossaries of IBM terms in multiple languages is available on the IBM Terminology Web site at: http://www.ibm.com/ibm/terminology/index.html
- National Language Support information; available at the IBM Publications Center at: http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi
- *SQL Reference for Cross-Platform Development*; available at the following developerWorks® site: http://www.ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html

The following information resources are not published by IBM but can be useful to users of DB2 for z/OS and related products:

- Database design topics:
  - *DB2 for z/OS and OS/390® Development for Performance Volume I*, by Gabrielle Wiorkowski, Gabrielle & Associates, ISBN 0-96684-605-2
  - *DB2 for z/OS and OS/390 Development for Performance Volume II*, by Gabrielle Wiorkowski, Gabrielle & Associates, ISBN 0-96684-606-0
  - *Handbook of Relational Database Design*, by C. Fleming and B. Von Halle, Addison Wesley, ISBN 0-20111-434-8
- Distributed Relational Database Architecture™ (DRDA®) specifications; http://www.opengroup.org
- Domain Name System: *DNS and BIND*, Third Edition, Paul Albitz and Cricket Liu, O'Reilly, ISBN 0-59600-158-4
- Microsoft Open Database Connectivity (ODBC) information; http://msdn.microsoft.com/library/
- Unicode information; http://www.unicode.org

Index

A
application programming skills 10
auditing 33
autonomic reoptimization 25
Availability enhancements
  ALTER STOGROUP support for SMS storage classes 23
  ALTER TABLESPACE 23
  CLONE tables 21
  column renaming 23
  CREATE STOGROUP support for SMS storage classes 23
  EARLY code, modify without IPL 23
  faster replacement of tables 21
  index logging 23
  index renaming 23
  online REBUILD INDEX 22
  online REORG with no BUILD2 phase 21
  SMS storage classes, support for 23
  universal table space
    partition-by-growth table space 21
    range-partitioned table space 21

B
BIGINT data type and function 5
BINARY data type and function 5

C
CD-ROM, books on 35
CLONE tables 21
COLLATION_KEY function 8
column, renaming 23
consistency across DB2 family 10
  large object support, enhancements to 10

D
data caching, in-memory 25
data insert performance, improved 31
database, automatic creation of 9
DB2 books online 35
DB2 Information Center for z/OS solutions 35
DB2 Optimization Expert for z/OS 28
DB2 QMF
  DB2 QMF for TSO/CICS 15
  DB2 QMF for WebSphere 15
  DB2 QMF for Workstation 15
  DB2 QMF High Performance Option 15
  packaging options 15
DB2 tools 15
DB2 Tools 17
  DB2 Optimization Expert for z/OS 28
  Optimization Service Center for DB2 for z/OS 27
DECFLOAT built-in data type 13

E
EARLY code, modify without IPL 23
EXCEPT keyword in a subselect 6

F
FETCH FIRST n ROWS ONLY 25
FETCH statement
  use for LOB data 10
  use for XML data 10
file reference variables 6

G
global query optimization 25

H
histogram statistics 25

I
index compression 29
index key randomization 29
index on expression 29
index on expression, creating 9
index page size, larger 29
index, renaming 23
information on demand, leveraging your enterprise 3
INSTEAD OF triggers 5
INTERSECT keyword in a fullselect 6

L
large object enhancements
  FETCH statement for LOB and XML data 10
  LOB file reference variables 10
large object support, enhancements to 10
large objects
  elimination of LOB locks for increased performance 12
  optimization of data retrieval for small and medium LOBs 12
large objects, performance enhancements for 12
large objects, utilities enhancements for
  CHECK DATA utility 11
  CHECK LOB utility 11
  LOB handling for LOAD and UNLOAD utilities 11
  REORG of LOB table spaces 11
library
  online 35
LOB data, FETCH statement for 10
LOB file reference variables 10
logging, improved performance of 30

M
MERGE statement 14

N
network trusted contexts 33
new function in DB2 V9.1 for z/OS, overview 1
not logging table spaces, expanded support 7

© Copyright IBM Corp. 2007

notices, legal 37

O

OLAP specifications
  DENSE_RANK 8
  RANK 8
  ROW_NUMBER 8
online books 35
online REBUILD INDEX 22
online REORG with no BUILD2 phase 21
optimistic currency control 14
Optimization Service Center for DB2 for z/OS 27
ORDER BY in subselects and fullselects 25

P

page-range screening 25
partition-by-growth table space 21
peformance improvements
  query optimization techniques 25
performance improvements 25
  CPU processing time for utilities, reduction in 25
  CPU processing time reduction for utilities 25
  data insert improvements 31
  DB2 Optimization Expert for z/OS 28
  histogram statistics 25
  logging 30
  Optimization Service Center for DB2 for z/OS 27
  query optimization management 27
  query performance
    index compression 29
    index key randomization 29
    index on expression 29
    index page size, larger 29
  sequential key insert, relief for 30
  SQL optimization improvements 25
  varying-length rows 30

Q

query optimization management 27
query optimization techniques
  autonomic reoptimization 25
  FETCH FIRST n ROWS ONLY 25
  global query optimization 25
  in-memory data caching 25
  ORDER BY in subselects and fullselects 25
  page-range screening 25

R

range-partitioned table space 21
REBUILD INDEX, online 22
regulatory compliance 33
  auditing, improved 33
  INSTEAD OF triggers 34
  network trusted contexts 33
  roles 33
  Secure Socket Layer protocol 33
  SSL protocol 33
  trace filtering 33
RENAME COLUMN clause 23
REORG, online with no BUILD2 phase 21
roles 33

S

Secure Socket Layer protocol 33
SELECT FROM DELETE function 5
SELECT FROM MERGE statement 14
SELECT FROM UPDATE function
  SQL consistency 5
sequential key insert, relief for 30
softcopy publications 35
SQL built-in data types
  DECFLOAT 13
  VARBINARY 13
SQL compatibility and leadership 4
SQL consistency
  automatic creation of a database 9
  automatic creation of a table space 9
  automatic creation of all system-required objects 9
  BIGINT data type and function 5
  BINARY data type and function 5
  COLLATION_KEY function 8
  EXCEPT keyword in a subselect 6
  expanded support for not logging table spaces 7
  file reference variables 6
  index on expression 9
  INSTEAD OF triggers 5
  INTERSECT keyword in a fullselect 6
  native support for SQL procedures 7
  OLAP specifications
    DENSE_RANK 8
    RANK 8
    ROW_NUMBER 8
  SQL procedures, nested compound statements in 7
SQL consistency improvements 5
SQL leadership, DB2 family firsts
  DECFLOAT built-in data type 13
  MERGE statement 14
  optimistic currency control 14
  SELECT FROM MERGE statement 14
  update detection 14
  VARBINARY data type 13
SQL leadership, family firsts
  TRUNCATE TABLE statement 13
SQL procedures, native support 7
SQL procedures, nested compound statements in 7
SSL protocol 33
system-required objects, automatic creation of 9

T

table space, automatic creation of 9
table space, partition-by-growth 21
table space, range-partitioned 21
table space, universal 21
trace filtering 33
TRUNCATE TABLE statement 13

U

universal table space 21
update detection 14
utilities enhancements for large objects
  CHECK DATA utility 11
  CHECK LOB utility 11
  LOAD and UNLOAD utilities, LOB handling for 11
  REORG of LOB table spaces 11
utilities, CPU reduction time in 25

V

VARBINARY data type 13
varying-length rows, improved performance for 30

X

XML
  application development 3
  database administration support 3
  integration of XML data and relational data 3
  performance benefits 3
  XML document retrieval 3
  XML document storage 3
XML data, FETCH statement for 10

Program Number: 5635-DB2

Printed in USA

GC18-9856-00