

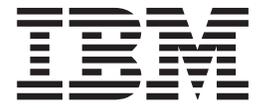
IBM® DB2 Universal Database™



Glossary

Version 8.2

IBM[®] DB2 Universal Database[™]



Glossary

Version 8.2

Before using this information and the product it supports, be sure to read the general information under *Notices*.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1993, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

DB2 Glossary	1	Release notes	113
DB2 Universal Database technical information	97	Printing DB2 books from PDF files	114
DB2 documentation and help	97	Ordering printed DB2 books	114
7 DB2 documentation updates.	97	Invoking contextual help from a DB2 tool	115
DB2 Information Center	98	7 Invoking message help from the command line processor	116
7 DB2 Information Center installation scenarios	99	7 Invoking command help from the command line processor	117
7 Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)	102	7 Invoking SQL state help from the command line processor	117
7 Installing the DB2 Information Center using the DB2 Setup wizard (Windows)	104	DB2 tutorials	117
7 Invoking the DB2 Information Center	106	DB2 troubleshooting information	118
7 Updating the DB2 Information Center installed on your computer or intranet server	107	Accessibility	119
7 Displaying topics in your preferred language in the DB2 Information Center	108	Keyboard input and navigation	119
DB2 PDF and printed documentation	109	Accessible display	119
7 Core DB2 information	109	Compatibility with assistive technologies	120
Administration information	109	Accessible documentation	120
Application development information	110	7 Dotted decimal syntax diagrams	120
Business intelligence information	111	7 Common Criteria certification of DB2 Universal	
DB2 Connect information	111	7 Database products.	122
Getting started information.	112	Notices	125
Tutorial information	112	Trademarks	127
Optional component information	112	Contacting IBM	129
		Product information	129

DB2 Glossary

Special Characters

7 **.NET Framework.** A Microsoft application development environment that consists of the common language runtime
7 and .NET Framework class library that is designed to provide a consistent programming environment for developing
7 and integrating code pieces. See also “common language runtime” on page 14.

A

abend. See “abnormal end of task.”

abend reason code. A 4-byte hexadecimal code that uniquely identifies a problem with DB2 Universal Database for z/OS and OS/390.

| **abnormal end of task (abend).** The termination of a task, job, or subsystem because of an error condition that
| recovery facilities cannot resolve during execution.

7 **abnormal termination.** (1) A system failure or operator action that causes a job to end unsuccessfully. (2) An exit
7 that is not under program control, such as a trap or a segmentation violation.

| **absolute path.** The full path name of an object. Absolute path names begin at the highest level, or root directory
| (which is identified by the forward slash (/) or backward slash (\) character).

| **access function.** A user-provided function that converts the data type of text stored in a column to a type that can
| be processed by the DB2 Net Search Extender.

access method services. A facility that is used to define and reproduce VSAM key-sequenced data sets.

| **access path.** The method that is selected by the database manager for retrieving data from a specific table. For
| example, an access path can involve the use of an index, a sequential scan, or a combination of the two.

access plan. The set of access paths that are selected by the optimizer to evaluate a particular SQL statement. The access plan specifies the order of operations to resolve the execution plan, the implementation methods (such as JOIN), and the access path for each table referenced in the statement.

| **access token.** (1) In DB2 Data Links Manager, an encrypted key that is assigned by the database manager and that
7 must be generated to access a file under the control of the Data Links Manager. (2) An object that contains security
7 information for a process or thread, including the identity and privileges of the user account that is associated with
7 the process or thread.

accounting string. User-defined accounting information that is sent to DRDA[®] servers by DB2 Connect. This information can be specified from the client workstation using the SQLESACT API or the DB2ACCOUNT environment variable, or the DB2 Connect workstation using the DFT_ACCOUNT_STR database manager configuration parameter.

7 **active log.** (1) The primary and secondary log files that are currently needed for recovery and rollback. (2) The
7 portion of the DB2 Universal Database for z/OS and OS/390 log to which log records are written as they are
| generated. The active log always contains the most recent log records. See also “archive log” on page 4.

7 **Activity Monitor.** An DB2 Universal Database administration tool that provides a set of predefined reports to assist
7 a database administrator in monitoring application performance and concurrency, resource consumption, and SQL
7 statement usage of a database or database partition. The tool also provides recommendations to help a database
7 administrator to diagnose the cause of database performance problems and to tune queries for optimal use of
7 database resources.

7 **address space.** (1) The actual memory that is used by an active program. See also “buffer pool” on page 9. (2) In
7 DB2 Universal Database for z/OS and OS/390, a range of virtual storage pages that is identified by a number (ASID)
| and a collection of segment and page tables that map the virtual pages to real pages of the computer’s memory.

Glossary

- | **adjacent nodes.** Two nodes connected by at least one path that connects no other nodes.
- | **administrative authority.** Either SYSADM authority level, which has full access to both instance and database
7 resources, or DBADM database authority, which has full access to database resources.
- | **administration notification log.** A list of national language messages that help the administrator to resolve minor
| issues. Also known as the DB2 notify log.
- | **administration notification message.** An alarm, error message, warning, attention message, or informational
7 message that is written by the database manager, replication programs, user applications, or the health monitor to a
7 notification file or event log.
- 7 **administration queue.** In Q replication and event publishing, a WebSphere MQ queue that is used for
7 communication between a Q Capture program and a Q Apply program or a user application. The administration
7 queue for each Q Capture program must be a local, persistent queue.
- | **administrative support table.** A table that is used by a DB2 extender to process user requests on image, audio, and
video objects. Some administrative support tables identify user tables and columns that are enabled for an extender.
Other administrative support tables contain attribute information about objects in enabled columns. Also called a
metadata table.
- | **administrator.** A person responsible for administrative tasks such as access authorization and content management.
| Administrators can also grant levels of authority to users. See also “user” on page 91.
- 7 **ADSM.** Deprecated name. See “Tivoli Storage Manager” on page 86.
- | **Advanced Peer-to-Peer Networking (APPN).** An extension to SNA that features distributed network control,
| dynamic definition of network resources, and automated resource registration and directory lookup. See also
| “Systems Network Architecture” on page 83.
- | **Advanced Program-to-Program Communication (APPC).** An implementation of the SNA LU 6.2 protocol that
| allows interconnected systems to communicate and share the processing of programs. See also “Common
| Programming Interface Communications” on page 14.
- 7 **after-image.** In SQL replication, the updated content of a source-table column that is recorded in a change data (CD)
7 table or in a database log or journal. Contrast with “before-image” on page 7.
- 7 **after trigger.** A trigger that is specified to be activated after the defined trigger event (insert, update, or delete
operation on the table that is specified in the trigger definition). See also “trigger” on page 88 and “before trigger” on
| page 7.
- 7 **after-value.** In Q replication, the updated content of a source-table column.
- | **agent.** (1) A separate process or thread that carries out all DB2 requests that are made by a particular client
| application. See also “warehouse agent” on page 93. (2) For z/OS and OS/390 environments, the structure that
| associates all processes that are involved in a unit of work. See also “system agent” on page 83, “coordinating agent”
| on page 17, and “allied agent” on page 3.
- | **agent site.** In the Data Warehouse Center, the location, defined by a single network host name, where a warehouse
| agent application is installed.
- 7 **agent thread.** In Q replication, one of the threads of the Q Apply program that receives transactions from a browser
7 thread and applies this data to target tables on the same server. One or more agent threads can exist for each browser
7 thread.
- 7 **aggregate function.** See “column function” on page 13.
- | **aggregate table.** In SQL replication, a read-only replication target table that contains aggregations of data from the
| source table. This data is based on SQL column functions such as MIN, MAX, SUM, or AVG.
- | **alert.** (1) A signal representing a state of an object (such as a database, table space, or instance). See “health monitor
| alert” on page 39.
| The types of alerts are listed in order of severity and include:
| • **attention**

- | An informational alert indicating that an object is in a non-normal state.
- | • **warning**
- | A non-critical condition that does not require immediate attention but might indicate a non-optimal system.
- | • **alarm**
- | A critical condition requiring immediate action.
- 7 (2) In replication, a notice that describes events and conditions in replication. The Replication Alert Monitor sends
- 7 alerts to an e-mail address or to a pager.
- 7 **alert condition.** In replication, a condition of the replication environment that causes the Replication Alert Monitor
- 7 to send alerts. Alert conditions can be triggered by status, by events, and by thresholds.
- | **alias.** An alternative name used to identify a table, view, database, or nickname. An alias can be used in SQL
- | statements to refer to a table or view in the same DB2 system or subsystem, or a remote DB2 system or subsystem.
- | **alias chain.** A series of table aliases that refer to one another in a sequential, nonrepeating fashion.
- | **allied address space.** An area of storage that is external to and connected to DB2 Universal Database for z/OS and
- | OS/390. An allied address space is capable of requesting DB2 Universal Database for z/OS and OS/390 services.
- | **allied agent.** Synonym for “allied thread.”
- | **allied thread.** A thread that originates at the local DB2 Universal Database for z/OS and OS/390 subsystem and can
- | access data at a remote DB2 Universal Database for z/OS and OS/390 subsystem. See also “thread” on page 85.
- | **allocated cursor.** A cursor that is defined for stored procedure result sets by using the SQL statement ALLOCATE
- | CURSOR.
- | **already verified.** An SNA LU 6.2 security option that allows DB2 Universal Database for z/OS and OS/390 to
- | provide the user’s verified authorization identifier when allocating a conversation. The user is not validated by the
- | partner subsystem.
- | **ambiguous cursor.** (1) A cursor is ambiguous if all of the following conditions are true:
- | • The SELECT statement is dynamically prepared.
- | • The SELECT statement does not include either the FOR READ ONLY clause or the FOR UPDATE clause.
- | • The LANGLEVEL bind option is SAA1.
- | • The cursor otherwise satisfies the conditions of a deletable cursor.
- | An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL; otherwise, it is considered
- | deletable. (2) In DB2 Universal Database for z/OS and OS/390, a database cursor that is not defined with the FOR
- | FETCH ONLY clause or the FOR UPDATE OF clause, is not defined on a read-only result table, is not the target of a
- | WHERE CURRENT clause on an SQL UPDATE or DELETE statement, and is in a plan or package that contains
- | either PREPARE or EXECUTE IMMEDIATE SQL statements. See also “unambiguous cursor” on page 89.
- | **American Standard Code for Information Interchange (ASCII).** An encoding scheme that is used to represent
- | character strings in many environments, typically on personal computers and UNIX systems. See also “EBCDIC” on
- | page 31 and “Unicode” on page 89.
- | **anti-join.** An answer set in which the returned rows do not meet the condition of the join predicate. See also “join”
- | on page 46.
- APF.** See “authorized program facility” on page 6.
- API.** See “application programming interface” on page 4.
- APPC.** See “Advanced Program-to-Program Communication ” on page 2.
- APPL.** A VTAM® network definition statement that is used to define DB2 Universal Database for z/OS and OS/390
- | to VTAM as an application program that uses SNA LU 6.2 protocols.
- | **application.** A program or set of programs that performs a task; some examples are payroll, inventory management,
- | and word processing applications.

Glossary

7 **Application Development Client.** An application development product that allows applications to be developed on
7 a client workstation to access remote database servers, including DB2 family databases, through the DB2 Connect
7 products.

1 **application-directed connections.** A connection that an application manages using the SQL CONNECT statement.
See also “system-directed connection” on page 83.

7 **application ID.** A unique string that is generated when the application connects to the database, or when DB2
7 Connect receives a request to connect to a Distributed Relational Database Architecture database. This ID is known
7 on both the client and the server and can be used to correlate the two parts of the application.

7 **application lock chain.** In the Activity Monitor, a graphical representation of lock-waiting relationships between a
7 selected application and other applications.

application name. The name of the application running on the client that identifies it to the database manager or
DB2 Connect. It is passed from the client to the server to establish the database connection.

application plan. The control structure that is produced during the bind process. DB2 Universal Database for z/OS
and OS/390 uses the application plan to process SQL statements that it encounters during statement execution.

7 **application process.** The unit to which the database manager allocates resources and locks. A process involves the
7 running of one or more programs. The running of an SQL statement is always associated with a process.

7 **application program.** A program that is used to connect and communicate with stations in a network, enabling
7 users to perform application-oriented activities.

1 **application programming interface (API).** A functional interface that allows an application program that is written
1 in a high-level language to use specific data or functions of the operating system or another program, such as a
1 database management system. In DB2 UDB, APIs enable most of the administrative functions from within an
1 application program.

1 **application requester.** The component on a remote system that generates DRDA requests for data on behalf of an
1 application. An application requester accesses a DB2 database server using the DRDA application-directed protocol.
1 See also “application server.”

7 **application server.** The target of a request from a remote application. In the DB2 environment, the application server
7 function is provided by the distributed data facility and is used to access DB2 data from remote applications. See also
7 “application requester.”

7 **apply.** In replication, to refresh or update a replication target table.

1 **Apply control server.** In SQL replication, a database or subsystem that contains the Apply control tables, which
1 store information about registered replication source tables and subscription sets. Contrast with “Apply server.”

1 **Apply cycle.** In SQL replication, the interval of time during which data is replicated from a source table to a target
1 table.

1 **Apply latency.** In SQL replication, an approximate measurement of the time that replication requires to complete
1 one cycle. See also “Capture latency” on page 10.

7 **Apply program.** In SQL replication, a program that is used to refresh or update a replication target table. Contrast
7 with “Capture program” on page 10 and “Capture trigger” on page 10.

7 **Apply qualifier.** In SQL replication, a case-sensitive character string that identifies replication subscription sets that
7 are unique to an instance of the Apply program.

1 **Apply server.** In SQL replication, a system where the Apply program is running. Contrast with “Apply control
1 server.”

APPN. See “Advanced Peer-to-Peer Networking” on page 2.

1 **archive log.** (1) The set of log files that is closed and is no longer needed for normal processing. These files are
1 retained for use in rollforward recovery. (2) The portion of the DB2 Universal Database for z/OS and OS/390 log that
7 contains log records that are copied from the active log. The archive log holds records that no longer fit on the active
7 log.

argument. A value passed to or returned from a function or procedure at run time.

| **ASCII.** See “American Standard Code for Information Interchange” on page 3.

7 **assembly.** In the .NET Framework, a file that contains intermediate language bytecode and that can be either a
7 library or an executable file. See also “intermediate language” on page 45.

| **AST.** See “automatic summary table” on page 6.

7 **asynchronous.** Pertaining to events that are not synchronized in time or do not occur in regular or predictable time
7 intervals. An input event is asynchronous if the program reads the data at an unspecified period of time after the
7 data is entered. See also “synchronous” on page 82.

asynchronous batched update. A process in which all changes to the source are recorded and applied to existing
target data at specified intervals. See also “asynchronous continuous update.”

asynchronous continuous update. A process in which all changes to the source are recorded and applied to existing
target data after being committed in the base table. See also “asynchronous batched update.”

| **asynchronous I/O.** The nonsequential processing of read and write requests across multiple disks.

7 **asynchronous mode.** In high availability disaster recovery (HADR), the synchronization mode in which the primary
7 database considers a transaction committed when it successfully submits the relevant log data to the network. The
7 primary database does not wait for an acknowledgement that the log data was received by the standby system. See
7 also “high availability disaster recovery” on page 40 and “peer state” on page 61.

7 **asynchronous replication.** In replication, the process of copying data from a source table to a target table outside
7 the scope of the original transaction that updated the source table. Contrast with “synchronous replication” on page
7 82.

7 **atomicity.** A technique within computer programming whereby either a group of statements are run as if a single
7 operation or none of the statements are run.

| **attach.** To remotely access objects at the instance level.

attachment facility. An interface between DB2 Universal Database for z/OS and OS/390 and TSO, IMS™, CICS, or
batch address spaces. An attachment facility allows application programs to access DB2 Universal Database for z/OS
and OS/390.

| **attachment relationship type.** In the Information Catalog Center, the relationship type that is used to attach
| comments to other objects. Comments can contain additional information about the object that they are attached to.
| See also “relationship type” on page 69.

7 **attribute.** A characteristic or trait of an entity that describes the entity; for example, the telephone number of an
7 employee is one of that employee’s attributes.

7 **audit.** To record information after the detection of monitored data access by applications or individuals.

| **audit facility.** A utility that generates a trail of audit records for a series of predefined and monitored database
events.

| **audit log file.** Location of audit records generated from the audit facility.

| **audit trail.** Data in the form of a logical path that links a sequence of events. An audit trail traces the transactions
| that affect the contents of a record.

| **authentication.** The process by which a system verifies a user’s identity. User authentication is completed by a
security facility outside of DB2 Universal Database, often part of the operating system or a separate product.

7 **authentication type.** One of a finite number of predefined methods for an instance or subsystem that is used to
7 determine how and where a user is authenticated before being permitted access to the instance or subsystem, or to
7 any objects found in that instance or subsystem.

| **authority.** See “authority level” on page 6.

Glossary

1 **authority level.** A user's access and ability to perform high-level database management operations such as
1 maintenance and utility operations. In addition to maintenance and utility operations, a user's authority level can
1 also implicitly include privileges on database objects. See also "load authority" on page 48, "system authority" on
1 page 83, "privilege" on page 63, "implicit privilege" on page 41, and "database authority" on page 20.

1 **authorization.** The DB2 Universal Database process by which data in the database catalog is used to obtain
1 information about the authenticated user, such as the database operations that the user can perform and which data
1 objects the user can access. See also "privilege" on page 63, "database authority" on page 20, and "authority level."

7 **authorization ID.** A character string in a statement that can be verified for connection to DB2 Universal Database
7 and to which a set of privileges is applied. An authorization identifier can represent an individual, an organizational
7 group, or a function, and is used by the database manager for authorization checking and as an implicit qualifier for
7 the names of objects such as tables, views, and indexes.

1 **authorization token.** (1) A token associated with a transaction. (2) For DB2 Universal Database for z/OS and
1 OS/390, the correlation ID. (3) For DB2 Universal Database for iSeries, the job name of the job that caused a
1 transaction.

authorized program facility (APF). In DB2 Universal Database for z/OS and OS/390, a facility that permits the
identification of programs that are authorized to use restricted functions.

autocommit. To automatically commit the current unit of work after each SQL statement.

1 **automatic configuration parameters.** A set of configuration parameters whose values can be changed automatically
by the database manager to reflect the current resource utilization.

7 **automatic load.** In Q replication, a load process in which the Q Apply program loads data into a target table. The
7 user can specify a load utility or let the Q Apply program choose the best available utility. See also "manual load" on
7 page 52.

7 **automatic maintenance.** A process by which DB2 Universal Database uses user-defined objectives to identify and
7 run required maintenance activities during the next available maintenance window. See also "maintenance window"
7 on page 52.

7 **automatic query rewrite.** A process that examines an SQL statement that refers to one or more base tables, and, if
7 appropriate, rewrites the query so that it performs better. This process can also determine whether to rewrite a query
7 so that it refers to one or more materialized query tables that are derived from the source tables.

1 **automatic rebind.** A process by which SQL statements are bound automatically (without a user issuing a BIND
1 command) when an application process begins execution and the bound application plan or package that it requires
1 is not valid. See also "bind" on page 8 and "rebind" on page 67.

1 **automatic summary table (AST).** A summary table defined such that changes made to the underlying tables are
1 cascaded to the summary table immediately and without the need for a REFRESH TABLE statement. See also
1 "summary table" on page 82 and "materialized query table" on page 52.

7 **autonomic.** Pertaining to an on-demand operating environment that responds automatically to problems, security
7 threats, and system failures.

7 **autonomic computing.** A computing environment with the ability to manage itself and dynamically adapt to change
7 in accordance with business policies and objectives. This self-managing environment can perform such activities
7 based on situations that it observes or senses in the IT environment rather than requiring IT professionals to initiate
7 the task. Autonomic computing systems have four key properties: self-configuring, self-healing, self-optimizing, and
7 self-protecting.

auxiliary index. In DB2 Universal Database for z/OS and OS/390, an index on an auxiliary table in which each
index entry refers to a LOB. See also "auxiliary table."

auxiliary table. A table that stores columns outside the table in which they are defined. See also "base table" on
page 7.

B

| **backout.** The process of undoing uncommitted changes that an application process has made. A backout might be necessary in the event of a failure on the part of an application process, or as a result of a deadlock situation. See also “rollback” on page 72.

| **backout free interval.** A set of log records that are not compensated if the transaction aborts. See also “backout.”

| **backup.** A copy of a database or table space that can be stored on a different medium and used to restore the database or table space in the event of failure or damage to the original.

backup pending. The state of a database or table space that prevents an operation from being performed until the database or table space is backed up.

backward log recovery. The fourth and final phase of restart processing during which DB2 Universal Database for z/OS and OS/390 scans the log in a backward direction to apply UNDO log records for all aborted changes.

| **base aggregate table.** In SQL replication, a type of replication target table that contains data that is aggregated from a replication source table. Contrast with “change aggregate table” on page 11.

7 **base table.** (1) A table that is created with the CREATE TABLE statement and is used to hold persistent user data. Such a table has both its description and data stored in the database. (2) In DB2 Universal Database for z/OS and OS/390, a table that contains a LOB column definition is also called a base table. The actual LOB column data is not stored with this base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns. See also “declared temporary table” on page 26, “auxiliary table” on page 6, “view” on page 92, “result table” on page 72, and “temporary table” on page 85.

| **base table space.** In DB2 Universal Database for z/OS and OS/390, a table space that contains base tables.

| **basic conversation.** An SNA LU 6.2 conversation between two transaction programs that uses the APPC basic conversation API. See also “mapped conversation” on page 52.

basic predicate. A predicate that compares two values.

basic sequential access method (BSAM). An access method that DB2 Universal Database for z/OS and OS/390 uses for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device. See also “queued sequential access method ” on page 66.

7 **before-image.** In SQL replication, the content of a replication source-table column before it is updated by a transaction. The content is recorded in a change data (CD) table or in a database log or journal. Contrast with “after-image” on page 2. See also “before-value.”

| **before trigger.** A trigger that is specified to be activated before the defined trigger event (insert, update, or delete operation on the table that is specified in the trigger definition). See also “trigger” on page 88 and “after trigger” on page 2.

7 **before-value.** In Q replication, the content of a replication source-table column before it is updated by a transaction.

7 **bidirectional replication.** In Q replication, a replication configuration in which changes that are made to one copy of a table are replicated to a second copy of that table. Changes that are made to the second copy are replicated back to the first copy.

7 **big endian.** A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first.

binary integer. A basic data type that can be further classified as small integer or large integer.

7 **binary large object (BLOB).** A data type that contains a sequence of bytes that can range in size from 0 bytes to 2 gigabytes less 1 byte. This string does not have an associated code page and character set. BLOBs can contain image, audio, and video data. See also “character large object” on page 11 and “double-byte character large object” on page 30.

| **binary string.** A sequence of bytes that is not associated with a CCSID. For example, the BLOB data type is a binary string. See also “coded character set identifier” on page 13.

Glossary

bind. To convert the output from the SQL compiler to a usable control structure, such as an access plan, application plan, or package. During the bind process, access paths to the data are selected and some authorization checking is performed. See also “rebind” on page 67, “automatic rebind” on page 6, “dynamic bind” on page 30, “incremental bind” on page 42, “static bind” on page 80.

| **bind file.** A file that is produced by the precompiler when the PRECOMPILE command or the respective API is used with the BINDFILE option.

bit data. Data with character type CHAR or VARCHAR that is not associated with a coded character set and therefore is never converted.

BLOB. See “binary large object” on page 7.

| **block.** (1) A string of data elements that is recorded or transmitted as a unit. (2) A set of contiguous data pages in a buffer pool. (3) A set of consecutive pages on disk.

| **block based I/O.** A database manager method of reading contiguous data pages from disk into contiguous portions of memory. See also “scattered read” on page 74.

7 **block factor.** See “block size.”

| **block fetch.** A function of DB2 Universal Database that retrieves (or fetches) a large set of rows together. Using block fetch can significantly reduce the number of messages that are sent across the network. Block fetch applies only to cursors that do not update data.

| **block identifier (BID).** An entry that is stored along with a key value in the leaf node of a block index. This identifier references a particular block in a multidimensional clustering table.

| **block index.** An index that is structured in the same manner as a traditional record identifier (RID) index, except that at the leaf level, keys point to a block identifier (BID) instead of an RID.

7 **blocking.** An option that allows caching of multiple rows of information by the communications subsystem so that each FETCH statement does not require the transmission of one row for each request across the network. This option is recommended when using SQL replication. See also “block fetch.”

| **block locks.** The locking of a block within a multidimensional clustering environment.

| **block map.** A bitmap that contains an array of block states, one for each block in the multidimensional clustering table. Each entry has eight bits, four of which are used:

- | • In use: set to 1 if the block is considered part of the table; 0 otherwise (that is, it is free).
- | • Load: set to 1 for newly loaded blocks; reset to 0 when the load utility completes.
- | • Constraint pending: set to 1 for newly loaded blocks; reset to 0 after constraints are checked.
- | • Refresh pending: set to 1 for newly loaded blocks; reset to 0 after automated summary table maintenance is completed.

| **block size.** Specifies the number of pages in a block. It is equal to the extent size. Also known as block factor.

| **bootstrap data set (BSDS).** A VSAM data set that contains name and status information for DB2 Universal Database for z/OS and OS/390, as well as relative-byte address-range specifications for all active and archive log data sets. It also contains passwords for the DB2 Universal Database for z/OS and OS/390 directory and catalog, and lists of conditional restart and checkpoint records.

| **broadcast join.** A join in which all partitions of a table are sent to all database partitions.

| **browse.** To view information catalog objects that are grouped by subject. Contrast with *search*.

| **browser.** (1) A function in DB2 Net Search Extender that enables you to display text on a computer screen. (2) A program that lets users look at data but not change it.

7 **browser thread.** In Q replication, a Q Apply program thread that gets messages from a receive queue and passes the messages to one or more agent threads to be applied to targets.

BSAM. See “basic sequential access method” on page 7.

BSDS. See “bootstrap data set” on page 8.

7 **buffer manipulator.** A process that is used in backup and restore operations to read from or write to the database.

1 **buffer pool.** An area of memory into which data pages are read, modified, and held during processing.

7 **built-in function.** A strongly typed, high-performance function that is integral to DB2 Universal Database. A built-in
7 function can be referenced in SQL statements anywhere that an expression is valid. See also “function” on page 36,
7 “sourced function” on page 77, “SQL function” on page 78, “external function” on page 34, and “user-defined
7 function” on page 91.

1 **business dimension.** A category of data, such as products or time periods, that an organization might want to
1 analyze. See also “dimension” on page 28 and “multidimensional analysis” on page 54.

1 **business metadata.** Data that describes information assets in business terms. Business metadata is stored in the
1 information catalog and accessed by users to find and understand the information that they need. For example,
1 business metadata for a program would contain a description of what the program does and which tables it uses. See
1 also “technical metadata” on page 85.

1 **business name.** In the Data Warehouse Center, a descriptive name that can be associated with an object that also
1 has a physical name. The object types that can have business names are tables, files, columns or fields. The business
1 name can be used in a search. It is also passed to end-user tools through the warehouse metadata interchange
1 facilities.

2 **bypass.** To allow a query to run without being managed by Query Patroller.

7 **byte reversal.** A technique in which numeric data is stored with the least significant byte first. The least significant
7 byte is the lowest byte in a number, located at the far right of a string.

C

cache. A buffer that contains frequently accessed instructions and data; it is used to reduce access time.

7 **cache structure.** A coupling facility structure that stores data that can be available to all members of a Parallel
7 Sysplex®. A DB2 Universal Database for z/OS and OS/390 data sharing group uses cache structures as a group
1 buffer pool.

7 **cache table.** In a federated system, a logical table object that is used to cache data from a data source table. A cache
7 table is comprised of a nickname that identifies the data source table, one or more materialized query tables, and a
7 schedule for replicating the data in each materialized query table.

1 **caching.** The process of storing frequently used results from a request to memory for quick retrieval, until it is time
1 to refresh the information. DB2 Universal Database provides many forms of caching, such as directory caching,
1 package caching, file system caching, and LDAP caching.

CAF. See “call attachment facility.”

7 **call.** To invoke a stored procedure by using the SQL CALL statement.

call attachment facility (CAF). A DB2 Universal Database for z/OS and OS/390 attachment facility for application
programs that run in TSO or MVS batch. The CAF is an alternative to the DSN command processor and provides
greater control over the execution environment.

7 **call level interface (CLI).** An API for database access that provides a standard set of functions to process SQL
7 statements and related services at run time. Contrast with “embedded SQL” on page 31.

7 **capture.** (1) In SQL replication, to gather changes from a source database and store them for replication to a target
7 database. These changes can come from the DB2 log or journal or from source transactions in a non-DB2 relational
7 database. (2) In Q replication, to gather changes from a source database and store them in a queue for replication to a
7 target database. (3) In event publishing, to gather changes from a source database and send them in XML format
7 through a queue to a user application.

Glossary

| **Capture control server.** (1) In SQL replication, a database or subsystem that contains the Capture control tables, which store information about registered replication source tables. (2) A system where the Capture program is running.

| **Capture latency.** In SQL replication, an approximate measurement of how recently the Capture program committed data to a CD table. See also "Apply latency" on page 4.

7 **Capture program.** In SQL replication, a program that reads database log or journal records to capture changes that are made to DB2 Universal Database source tables and store them in staging tables. Contrast with "Apply program" on page 4, "Capture trigger," and "Q Capture program" on page 65.

| **Capture schema.** In SQL replication, a name that identifies the control tables that are used by a particular instance of the Capture program.

7 **Capture trigger.** In SQL replication, a mechanism that captures delete, update, or insert operations that are performed on non-DB2 source tables. Contrast with "Capture program" and "Apply program" on page 4.

cardinality. The number of rows in a database table.

1 **cascade.** In the Data Warehouse Center, to run a sequence of events. When a step cascades to another step, the steps run sequentially or concurrently. A step can also cascade to a program, which runs after the step finishes running.

1 **cascade delete.** The way in which DB2 Universal Database enforces referential constraints when it deletes all descendent rows of a deleted parent row.

7 **cascade rejection.** In SQL replication, the process of rejecting a replication transaction because it is associated with a transaction that had a conflict detected and was itself rejected.

7 **cascading trigger.** See "trigger cascading" on page 88.

| **CASE expression.** An expression that allows another expression to be selected based on the evaluation of one or more conditions.

| **case-insensitive search.** A search result without consideration of the case of the string being searched.

| **cast function.** A function that is used to convert instances of a source data type into instances of a different target data type. In general, a cast function has the name of the target data type and has one single argument whose type is the source data type. Its return type is the target data type.

1 **catalog.** A set of tables and views that are maintained by the database manager. These tables and views contain information about the database, such as descriptions of tables, views, and indexes. See "information catalog" on page 43, "database catalog" on page 20, and "RDBMS catalog" on page 67.

catalog node. See "catalog partition."

| **catalog partition.** In a partitioned database environment, the database partition where the catalog tables for the database are stored. Each database in a partitioned database environment can have its catalog partition on a different database partition server. The catalog partition for a database is automatically created on the database partition server where the CREATE DATABASE command is run.

7 **catalog table.** A table that is automatically created in the DB2 Universal Database catalog when the database is created. These tables contain information about the database and its objects, such as the definitions of database objects and security information about the authority that users have on these objects.

7 **catalog view.** (1) A SYSCAT or SYSSTAT view on the catalog table. (2) One of several views created when DB2 Net Search Extender is enabled for a database. Catalog views contain information about text search configurations and indexes in the database.

7 **catchup state.** In high availability disaster recovery (HADR), a state in which the standby database might not have applied all logged operations that occurred at the primary database. In this state, the standby database retrieves and applies previously generated log data to synchronize with the primary database. There are two types of catchup states: local and remote. See also "local catchup state" on page 49 and "remote catchup state" on page 70.

CCD table. See "consistent-change-data table" on page 16.

CCSID. See “coded character set identifier” on page 13.

CDB. See “communications database” on page 14.

CDRA. See “Character Data Representation Architecture.”

CD table. See “change-data table.”

| **CelDial sample catalog.** A sample information catalog (ICCSAMP) available when you install the Information
| Catalog Center. An administrator initializes the catalog, and users can use the sample data to become familiar with
| the Information Catalog Center.

| **cell.** A unique combination of dimension values. Physically, a cell is made up of blocks of pages whose records all
| share the same values for each clustering column.

7 **central processor complex (CPC).** In a z/OS or OS/390 environment, a physical collection of hardware (such as an
7 ES/3090 system) that consists of main storage, one or more central processors, timers, and channels.

| **CFRM.** See “Coupling Facility Resource Management” on page 18.

CFRM policy. In DB2 Universal Database for z/OS and OS/390, a declaration by an MVS administrator regarding
the allocation rules for a coupling facility structure.

7 **change aggregate table.** In SQL replication, a type of replication target table that contains data aggregations that are
7 based on the contents of a CD table. Contrast with “base aggregate table” on page 7.

7 **change-capture replication.** In replication, the process of capturing changes that are made to a replication source
7 table and applying them to a replication target table. Contrast with “full refresh” on page 36.

7 **change-data table (CD table).** In replication, a replication table at the Capture control server that contains changed
7 data for a replication source table.

7 **Character Data Representation Architecture (CDRA).** An IBM architecture that defines a set of identifiers,
7 resources, services, and conventions to achieve consistent representation, processing, and interchange of graphic
7 character data in heterogeneous environments.

| **character conversion.** The process of changing data from one character coding representation to another.

7 **character large object (CLOB).** A data type that contains a sequence of characters (single-byte, multi-byte, or both)
7 that can range in size from 0 bytes to 2 gigabytes less 1 byte. In general, character large object values are used
7 whenever a character string might exceed the limits of the VARCHAR type. Also called character large object string.
7 See also “binary large object” on page 7 and “double-byte character large object” on page 30.

| **character set.** A defined set of characters. For example, 26 nonaccented letters A through Z.

| **character string.** A sequence of bytes that represent bit data, single-byte characters, or a mixture of single-byte and
| multibyte-byte characters.

| **character string delimiter.** The characters that are used to enclose character strings in delimited ASCII files that are
| imported or exported. See also “delimiter” on page 27.

2 **charge-back account.** An account used for tracking computing resource usage by business departments or projects.
2 Examples of charge-back accounts are “Marketing Department” or “Summer catalog project”.

CHECK clause. In SQL, an extension to the CREATE TABLE and ALTER TABLE statements that specifies a table
check constraint.

check condition. A restricted form of search condition that is used in check constraints.

7 **check constraint.** A rule that is associated with a table that specifies the values allowed in one or more columns of
7 every row in the table. A check constraint is optional and can be defined using the CREATE TABLE or the ALTER
7 TABLE statement. See also “constraint” on page 16, “unique constraint” on page 90, and “informational constraint”
7 on page 43.

Glossary

check integrity. The condition that exists when each row in a table conforms to the check constraints that are defined on that table. Maintaining check integrity requires DB2 to enforce table check constraints on operations that add or change data.

check pending. A state into which a table can be put where only limited activity is allowed on the table and constraints are not checked when the table is updated.

checkpoint. A point at which the database manager records internal status information on the log; the recovery process uses this information if the subsystem abnormally terminates.

child lock. In explicit hierarchical locking, a lock that is held on a table, a page, a row, or a large object. Each child lock has a parent lock. See also “parent lock” on page 60.

CI. See “control interval” on page 17.

CICS. An IBM licensed program that provides online transaction-processing services and management for business applications.

CICS attachment facility. A facility that provides a multithread connection to DB2 Universal Database to allow applications that run in the CICS environment to execute DB2 commands.

CIDF. See “control interval definition field” on page 17.

circular log. A database log in which records are overwritten if they are no longer needed by an active database. See also “database log” on page 20 and “archive log” on page 4.

claim. In DB2 Universal Database for z/OS and OS/390, a notification to the database manager that an object is being accessed. Claims prevent drains from occurring until the claim is released, which usually occurs at a commit point. See also “drain” on page 30.

claim class. In DB2 Universal Database for z/OS and OS/390, a specific type of object access that can be one of the following types: cursor stability, repeatable read, or write.

claim count. In DB2 Universal Database for z/OS and OS/390, a count of the number of agents that are accessing an object.

class of service. In DB2 Universal Database for z/OS and OS/390, a VTAM term for a list of routes through a network, arranged in an order of preference for their use.

class word. A single word that indicates the nature of a data attribute.

clause. In SQL, a distinct part of a statement, such as a SELECT clause or a WHERE clause.

clean block index. An index such that every record in a block that is referenced by the index has the same key value for that index. A dimension block index is a clean block index.

cleanse. (1) To ensure that all values in a data set are consistent and correctly recorded. (2) To transform the data extracted from operational systems to make it usable by the data warehouse.

CLI. See “call level interface” on page 9.

client. A system or process that is dependent on another system or process (usually called the server) to provide it with access to data, services, programs, or resources.

client profile. A profile that is used to configure clients using the Import function in the Configuration Assistant. It can contain database connection information, client settings, CLI or ODBC common parameters, and configuration data for local APPC or NetBIOS communication subsystems. See also “server profile” on page 76.

client reroute. A method that allows a client application, upon the loss of communication with a database server and the predefinition of an alternative server, to continue working with the original database server or the alternative server with only minimal interruption of the work.

CLIST. See “command list” on page 14.

CLOB. See “character large object” on page 11.

CLP. See “command line processor” on page 14.

CLPA. See “create link pack area” on page 19.

7 **CLR.** See “common language runtime” on page 14.

7 **clustered index.** An index whose sequence of key values closely corresponds to the sequence of rows that are stored
7 in a table. The degree to which this correspondence exists is measured by statistics that are used by the optimizer.

7 **clustering index.** An index that determines how rows are physically ordered (clustered) in a table space. If a
7 clustering index on a partitioned table is not a partitioning index, the rows are ordered in cluster sequence within
7 each data partition instead of spanning the partitions. Prior to Version 8 of DB2 Universal Database for z/OS, the
7 partitioning index was required to be the clustering index.

1 **clustering block index.** See “dimension block index” on page 28.

coded character set. A set of unambiguous rules that establishes a character set and the one-to-one relationships between the characters of the set and their coded representations.

coded character set identifier (CCSID). A number that includes an encoding scheme identifier, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic character representation.

1 **code page.** A set of assignments of characters to code points.

1 **code point.** A unique bit pattern that represents a character in a code page.

1 **code set.** International Organization for Standardization (ISO) term for code page. See “code page.”

1 **cold start.** (1) The process of starting a system or program by using an initial program load procedure. (2) A process
1 by which DB2 Universal Database for z/OS and OS/390 restarts without processing any log records. See also “warm
7 start” on page 93. (3) In replication, the process of starting the Capture program or the Q Capture program without
7 using restart information from prior operation of the program. During a cold start, the Capture program or Q
7 Capture program initiates a full refresh of the target tables. Contrast with “warm start” on page 93.

collating sequence. The sequence in which the characters are ordered for the purpose of sorting, merging, comparing, and processing indexed data sequentially.

collection. (1) In DB2 Universal Database for z/OS and OS/390, a group of packages that have the same qualifier.
1 (2) In the Information Catalog Center, a container for objects. A collection contains objects that the user has privileges
1 to see, similar to a personal folder of objects.

1 **collocated join.** The result of two tables being joined when the tables reside in a single-partition database partition
1 group in the same database partition; or they are in the same database partition group and have the same number of
1 partitioning columns, the columns are partition-compatible, and both tables use the same partitioning function, and
1 pairs of the corresponding partitioning key columns participate in the equijoin predicates. See also
1 “partition-compatible join” on page 60.

1 **column data.** The data store that is stored in a DB2 column. The type of data can be any data type that is supported
1 by DB2.

column distribution value. Statistics that describe the most frequent values of some column or the quantile values. These values are used in the DB2 optimizer to help determine the best access plan.

7 **column function.** A function that optionally accepts arguments and returns a single scalar value that is the result of
7 an evaluation over a set of like values, such as those in a column within a set of one or more rows. Also known as an
7 aggregate function. See also “function” on page 36, “scalar function” on page 74, “row function” on page 73, and
7 “table function” on page 84.

1 **column options.** In a federated system, parameters of the ALTER NICKNAME statement that describe the values in
1 certain columns of the data source object that a nickname references. This information is added to the global catalog
1 and used by the DB2 query optimizer to develop better access plans. Column options provide a way to tell the data
1 source wrapper to handle a column in a different way than it normally would.

Glossary

1 **come-from checking.** An SNA LU 6.2 security option that defines a list of authorization identifiers that are allowed
to connect to DB2 Universal Database for z/OS and OS/390 from a partner LU.

1 **command.** A way to start database administration functions to access and maintain the database manager. See also
1 “DB2 command” on page 24.

7 **command line processor (CLP).** A text-based interface for entering SQL statements and database manager
7 commands.

command list. A language that DB2 Universal Database for z/OS and OS/390 uses to perform TSO tasks.

command prefix. In DB2 Universal Database for z/OS and OS/390, a one- to eight-character command identifier.
The command prefix distinguishes the command as belonging to an application or subsystem rather than to DB2
Universal Database for z/OS and OS/390.

command recognition character (CRC). A character that permits an MVS console operator or an IMS subsystem
user to route DB2 commands to specific DB2 Universal Database for z/OS and OS/390 subsystems.

7 **command scope.** The breadth of the impact of a command in a data sharing group. In a data sharing environment,
7 a command can have a group scope or a member scope. See also “group scope” on page 39 and “member scope” on
7 page 52.

1 **comments object type.** An object type that annotates another object in the Information Catalog Center. For example,
1 you can attach a comment to a chart object that contains notes about the data in the chart. The comments object type
1 is predefined in the Information Catalog Center.

commit. The operation that ends a unit of work by releasing locks so that the database changes made by that unit
of work can be perceived by other processes. This operation makes the data changes permanent.

1 **commit point.** A point in time when data is considered to be consistent.

committed phase. The second phase of the multisite update process that requests all participants to commit the
effects of the logical unit of work.

common-index table. A DB2 table whose text columns share a common text index.

7 **common language runtime (CLR).** The run-time interpreter for all .NET Framework applications that interprets
7 compiled assemblies. See also “assembly” on page 5 and “intermediate language” on page 45.

7 **Common Programming Interface Communications (CPI-C).** A call-level interface that provides a consistent
7 application programming interface for applications that use program-to-program communications. The interface uses
7 LU 6.2 architecture to create a set of interprogram services that can establish and end a conversation, send and
7 receive data, exchange control information, and notify a partner program of errors.

common service area (CSA). In OS/390, a part of the common area that contains data areas that can be addressed
by all address spaces.

1 **common table expression.** An expression that defines a result table with a name (a qualified SQL identifier). The
1 expression can be specified as a table name in any FROM clause in the fullselect that follows the WITH clause. See
1 also “table expression” on page 84.

communications database (CDB). A set of tables in the DB2 Universal Database for z/OS and OS/390 catalog that
is used to establish conversations with remote database management systems.

1 **comparison operator.** Comparison operators are \neq (not less than), $<$ (less than), \leq (less than or equal to), \neq (not
equal to), $=$ (equal to), \geq (greater than or equal to), $>$ (greater than), and \neq (not greater than). See also “infix
operator” on page 43.

1 **compensation.** In a federated system, the ability of DB2 to process SQL that is not supported by a data source. DB2
1 will not push down a query fragment if the data source cannot process it, or if DB2 can process it faster than the
7 data source can. If the data source cannot process it, DB2 will process it instead. There are two basic ways that a
7 federated server compensates for the loss of functionality at the data source: it will simulate the data source function,
7 or it will move the set of data to the federated server and perform the function locally. See also “query optimizer” on
1 page 66 and “push-down processing” on page 65.

complete. A table attribute that indicates that the table contains a row for every primary key value of interest. As a result, a complete source table can be used to perform a refresh of a target table.

7 **complete CCD table.** In SQL replication, a CCD table that initially contains all of the rows from the replication
7 source table or view and any predicates from the source table or view. Contrast with “noncomplete CCD table” on
7 page 55 and “consistent-change-data table” on page 16.

| **compose.** In the XML Extender, to generate XML documents from relational data in an XML collection.

| **composite block index.** An index that contains only dimension key columns and is used to maintain the clustering
| of data over insert and update activity in a multidimensional clustering (MDC) table. See also “dimension block
| index” on page 28.

composite key. An ordered set of key columns of the same table.

compound SQL statement. A block of SQL statements that are executed in a single call to the application server.

compression dictionary. In DB2 Universal Database for z/OS and OS/390, the dictionary that controls the process
of compression and decompression. This dictionary is created from the data in the table space or table space
partition.

concurrency. The shared use of resources by multiple interactive users or application processes at the same time.

7 **condensed.** In SQL replication, a table attribute that indicates that the table contains current data rather than a
7 history of changes to the data. A condensed table includes no more than one row for each primary key value in the
table. As a result, a condensed table can be used to supply current information for a refresh.

7 **condensed CCD table.** In SQL replication, a CCD table that contains only the most current value for a row and has
7 only one row for each key value. Contrast with “noncomplete CCD table” on page 55. See also “consistent-change-
7 data table” on page 16.

| **condition.** A specification of either the criteria for selecting XML data, or the way to join the XML collection tables.

conditional restart. In DB2 Universal Database for z/OS and OS/390, a restart that is directed by a user-defined
conditional restart control record (CRCR).

7 **conditional restart control record (CRCR).** In DB2 Universal Database for z/OS and OS/390, a queue of records in
the bootstrap data set (BSDS) that is associated with a conditional restart of DB2 Universal Database. Each element in
the queue indicates the choices that were made when the record was created (through the change log inventory
7 utility, DSNJU003), and the progress of the restart operation it controls. See also “conditional restart.”

| **configurable configuration parameters.** A set of configuration parameters that contain information that can be
| changed. See also “configurable online configuration parameters” and “informational configuration parameter” on
| page 43.

| **configurable online configuration parameters.** A set of configuration parameters whose values can be changed
| while the database manager is running.

| **configuration file.** A file that contains the values that are specified for configuration parameters. There are two
| types of configuration files: the database manager configuration file for each DB2 Universal Database instance, and
| the database configuration file for each individual database.

| **configuration parameter.** A parameter whose value limits the resources that can be used by the database manager
| or database. Some configuration parameters are informational, and display characteristics about the environment that
| cannot be changed.

7 **conflict detection.** (1) The method that is used at run time to detect any read or write operations that access the
7 same table at the same time. This method might result in inconsistent or incorrect semantics. (2) In bidirectional,
7 peer-to-peer, and update-anywhere replication, the process of detecting constraint errors such as key constraints and
7 referential constraints, or the process of detecting whether the same row was updated by users or application
7 programs in both the source table and target table during the same replication cycle.

connection. (1) An association between an application process and an application server. (2) In data
communications, an association established between functional units for conveying information. (3) In SNA, the

Glossary

existence of a communication path between two partner LUs that allows information to be exchanged (for example, two DB2 Universal Database for z/OS and OS/390 subsystems that are connected and communicating by way of a conversation).

| **connection concentrator.** A process that allows applications to stay connected without any resources being
| consumed on the DB2 host server. Thousands of users can be active in applications, while only a few threads are
| active on the DB2 host server.

| **connection handle.** The data object containing information that is associated with a connection that is managed by
| DB2 ODBC. This information includes general status information, transaction status, and diagnostic information. See
| also “statement handle” on page 80.

connection ID. In DB2 Universal Database for z/OS and OS/390, an identifier that is supplied by the attachment
facility and that is associated with a specific address space connection.

| **connection pooling.** A process in which DB2 Connect drops the inbound connection with an application that
| requests disconnection, but keeps the outbound connection to the host in a pool. When a new application requests a
| connection, DB2 Connect uses one from the existing pool. Using the already-present connection reduces the overall
| connection time, as well as the high processor connect cost on the host.

| **consistency token.** A timestamp that is used to generate the unique identifier (version identifier in DB2 Universal
| Database for z/OS and OS/390) for an application.

7 **consistent-change-data table (CCD table).** In SQL replication, a type of replication target table that is used for
7 storing history, auditing data, or staging data. A CCD table can also be a replication source. See also “complete CCD
7 table” on page 15, “condensed CCD table” on page 15, “external CCD table” on page 33, “internal CCD table” on
7 page 45, “noncomplete CCD table” on page 55, and “noncondensed CCD table” on page 56.

constant. A language element that specifies an unchanging value. Constants are classified as string constants or
numeric constants. See also “variable” on page 92.

| **constraint.** A rule that limits the values that can be inserted, deleted, or updated in a table. See also “check
| constraint” on page 11, “informational constraint” on page 43, “referential constraint” on page 68, and “unique
| constraint” on page 90.

7 **contact.** A person whose ID is configured to receive e-mail or pager notifications of DB2 administration messages
7 that are written to the notification log. The definition for each contact contains the name and the e-mail or pager
7 address of the person to receive notifications, and is stored in the contact list of the system that is specified by the
7 CONTACT_HOST configuration parameter of the DB2 administration server. See also “orphaned contact” on page 58,
7 “administration notification log” on page 2, and “administration notification message” on page 2.

| **contact relationship type.** In the Information Catalog Center, the relationship type that is used to identify contacts.
| A contact relationship type provides more information about an object. Such information might include the person
| who created the information that the object represents or the department that is responsible for maintaining the
| information. See also “relationship type” on page 69.

| **container.** A physical storage location of the data. For example, a file, directory, or device. See “table space
| container” on page 84.

| **contains relationship type.** In the Information Catalog Center, the relationship type that is used to identify
| Information Catalog Center objects that contain other objects. For example, use the contains relationship type to
| denote an object with a parent role, meaning that object can contain other objects. You can also use the contains
| relationship type to denote an object with a child role, meaning an object that can be contained in another object. See
| also “relationship type” on page 69.

contention. A situation in which a transaction attempts to lock a row or table that is already locked.

| **contracting conversion.** A process that occurs when the length of a converted string is smaller than that of the
| source string. See also “expanding conversion” on page 32.

7 **Control Center.** The DB2 Universal Database graphical interface that lets you administer databases and perform a
7 variety of tasks including creating objects and monitoring performance. The Control Center shows database objects
7 (such as databases and tables) and their relationship to each other.

control interval. In VSAM, a fixed-length area of direct access storage in which VSAM stores records and creates distributed free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always includes an integral number of physical records.

control interval definition field (CIDF). In VSAM, a field located in the 4 bytes at the end of each control interval; it describes the free space, if any, in the control interval.

control metadata. In the Data Warehouse Center, information about changes to the warehouse, such as the date and time that a table is updated by the processing of a step.

7 **control message.** In Q replication, a message from a Q Apply program or a user application that asks a Q Capture
7 program to activate or deactivate a Q subscription or an XML publication, invalidate a send queue, or confirm that a
7 target table is loaded.

| **control point.** In APPN, a component of a node that manages resources of that node and optionally provides
| services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 node, a
| physical unit control point (PUCP) in a type 4 node, a network node control point (NNCP) in a type 2.1 (T2.1)
| network node, and an end node control point (ENCP) in a T2.1 end node. An SSCP and an NNCP can provide
| services to other nodes. See also “physical unit” on page 62 and “control point name.”

| **control point name.** A network-qualified name of a control point that consists of a network identifier qualifier that
| identifies the network to which the control point node belongs. See also “control point.”

control privilege. The authority to completely control an object, which includes the authority to access, drop, or alter an object, and the authority to extend or revoke privileges on the object to other users.

7 **control server.** In SQL replication, a database server that contains replication control tables for the Capture program,
7 Apply program, or Replication Alert Monitor. See also “Apply control server” on page 4, “Capture control server” on
7 page 10, “Q Apply server” on page 65, “Q Capture server” on page 65, and “Monitor control server” on page 53.

| **control table.** See “replication control table” on page 71.

conversation. In APPC, a connection between two transaction programs over a logical unit-logical unit (LU-to-LU) session that allows them to communicate with each other while processing a transaction.

conversational transaction. In APPC, two or more programs communicating using the services of logical units (LUs).

| **conversation security.** In APPC, a process that allows validation of a user identifier or group identifier and
| password before establishing a connection.

| **conversation security profile.** The set of user identifiers or group identifiers and passwords that are used by APPC
| for conversation security.

7 **coordinate.** A member of an ordered set of N numbers that identifies a position in N-dimensional space. For
7 example, in a two-dimensional map of the Earth, a position can be referenced by two coordinates. The first
7 coordinate identifies the latitude value of the position, and the second coordinate identifies the longitude value of the
7 position.

| **coordinating agent.** The agent that is started when the database manager receives a request from an application.
| The agent remains associated with the application during the life of the application. This agent initiates subagents
| that work for the application. See also “agent” on page 2 and “subagent” on page 81.

coordinator. In DB2 Universal Database for z/OS and OS/390, the system component that initiates the commit or rollback of a unit of work that includes work that is done on one or more other systems.

coordinator node. See “coordinator partition.”

| **coordinator partition.** The database partition server to which the application originally connected and on which the
| coordinating agent resides.

coordinator subsection. The subsection of an application that starts other subsections (if any) and returns results to the application.

Glossary

7 **copy pool.** A named set of SMS storage groups that contains data that is to be copied collectively. A copy pool is an
7 SMS construct that lets you define which storage groups are to be copied by using FlashCopy® functions. HSM
7 determines which volumes belong to a copy pool.

7 **copy target.** A named set of SMS storage groups that are to be used as containers for copy pool volume copies. A
7 copy target is an SMS construct that lets you define which storage groups are to be used as containers for volumes
7 that are copied by using FlashCopy functions.

7 **copy version.** A point-in-time FlashCopy copy that is managed by HSM. Each copy pool has a version parameter
7 that specifies how many copy versions are maintained on disk.

correlated columns. In SQL, a relationship between the value of one column and the value of another column.

correlated reference. A reference to a column of a table that is outside a subquery.

correlated subquery. (1) A subquery that contains a correlated reference to a column of a table that is outside the
| subquery. (2) In DB2 Universal Database for z/OS and OS/390, a subquery that is part of a WHERE or HAVING
| clause that is applied to a row or group of rows of a table or view that is named in an outer subselect statement.

| **correlation ID.** In DB2 Universal Database for z/OS and OS/390, an identifier that is associated with a specific
| thread. In TSO, it is either an authorization identifier or the job name.

correlation name. An identifier that designates a table or view within a single SQL statement. The name can be
defined in any FROM clause or in the first clause of an UPDATE or DELETE statement.

| **cost.** The estimated total resource usage that is necessary to run the access plan for a statement (or the elements of a
| statement). Cost is derived from a combination of processor cost (in number of instructions) and I/O (in numbers of
| seeks and page transfers).

| **cost category.** A category into which DB2 Universal Database for z/OS and OS/390 places cost estimates for SQL
| statements at the time the statement is bound.

| **counter.** A representation of information that is cumulative up until the sample is taken. The counter counts values
| that increase, such as the number of deadlocks. Counters are reset when you stop and restart an instance or database.
| See also “gauge” on page 37.

| **country/region code.** See “territory code” on page 85.

| **Coupling Facility Resource Management.** In a z/OS or OS/390 environment, a function that manages all of the
7 coupling facilities in a sysplex.

7 **coupling facility.** In a z/OS or OS/390 environment, a designated PR/SM™ LPAR logical partition that runs the
7 coupling facility control program and provides high-speed caching, list processing, and locking functions in a
7 sysplex.

CP. See “control point” on page 17.

CPC. See “central processing complex” on page 11.

CPI-C. See “Common Programming Interface Communications” on page 14.

CPI-C side information profile. In SNA, the profile that specifies the conversation characteristics to use when
allocating a conversation with a remote transaction program. The profile is used by local transaction programs that
communicate through CPI Communications. It specifies the partner LU name (the name of the connection profile that
contains the remote LU name), the mode name, and the remote transaction program name.

CP name. See “control point name” on page 17.

| **crash recovery.** The process of bringing a database back to a consistent and usable state after a failure. See also
| “version recovery” on page 92 and “rollforward recovery” on page 72.

CRC. See “command recognition character” on page 14.

CRCR. See “conditional restart control record” on page 15.

- | **create link pack area (CLPA).** An option that is used during initial program load to initialize the link pack pageable area.
- | **created temporary table.** In DB2 Universal Database for z/OS and OS/390, a table that holds temporary data and is defined with the SQL statement CREATE GLOBAL TEMPORARY TABLE. Information about created temporary tables is stored in the DB2 catalog, so this kind of table is persistent and can be shared across application processes. See “temporary table” on page 85. See also “declared temporary table” on page 26.
- 7 **cross-memory linkage.** In a z/OS or OS/390 environment, a method for invoking a program in a different address space. The invocation is synchronous with respect to the caller.
- cross-system coupling facility (XCF).** A component of OS/390 that provides functions to support cooperation between authorized programs running within a Sysplex.
- cross-system extended services (XES).** A set of OS/390 services that enable multiple instances of an application or subsystem, running on different systems in a Parallel Sysplex environment, to implement high-performance, high-availability data sharing by using a coupling facility.
- CS.** See “cursor stability.”
- CSA.** See “common service area” on page 14.
- cumulative backup.** See “incremental backup” on page 42.
- current data.** In DB2 Universal Database for z/OS and OS/390, data within a host structure that is current with (identical to) the data within the base table.
- | **current path.** An ordered list of schema names that is used in the resolution of unqualified references to functions and data types. In dynamic SQL, the current function path is found in the CURRENT PATH special register. In static SQL, it is defined in the FUNCPATH option for PREP and BIND commands.
- current status rebuild.** In DB2 Universal Database for z/OS and OS/390, the second phase of restart processing during which the status of the subsystem is reconstructed from information on the log.
- | **current SQL ID.** An identifier that, at a single point in time, holds the privileges that are exercised when certain dynamic SQL statements run. The current SQL ID can be a primary authorization ID or a secondary authorization ID.
- current working directory.** The default directory of a process from which all relative path names are resolved.
- cursor.** A named control structure that is used by an application program to point to a specific row within some ordered set of rows. The cursor is used to retrieve rows from a set.
- | **cursor blocking.** A technique that reduces overhead by retrieving a block of rows in a single operation. These rows are cached while they are processed.
- | **cursor sensitivity.** The degree to which database updates are visible to the subsequent FETCH statements in a cursor. A cursor can be sensitive to changes that are made with positioned UPDATE and DELETE statements that specify the name of the cursor. A cursor can also be sensitive to changes that are made with searched UPDATE or DELETE statements, or with cursors other than this cursor. These changes can be made by this application process or by another application process.
- cursor stability (CS).** An isolation level that locks any row accessed by a transaction of an application while the cursor is positioned on the row. The lock remains in effect until the next row is fetched or the transaction is terminated. If any data is changed in a row, the lock is held until the change is committed to the database. See also “read stability” on page 67, “repeatable read” on page 70, and “uncommitted read (UR)” on page 89.
- 7 **Customer Information Control System.** See “CICS” on page 12.
- | **cycle.** In DB2 Universal Database for z/OS and OS/390, a set of tables that can be ordered so that each table is a descendent of the one before it, and the first table is a descendent of the last table. For example, a self-referencing table is a cycle with a single member.
- | **cyclical referential constraint.** A table that is a dependent of, or descendent of, another table.

Glossary

D

DAD. See “document access definition” on page 29.

| **daemon.** A system process that provides a specific service to applications or users.

7 **DADX.** See “Document Access Definition Extension” on page 29.

DARI. See “Database Application Remote Interface.”

| **data area.** A memory area that is used by a program to hold information.

7 **database.** A collection of interrelated or independent data items that are stored together to serve one or more
7 applications. See also “relational database” on page 69.

| **database access thread.** In DB2 Universal Database for z/OS and OS/390, a thread that accesses data at the local
| subsystem on behalf of a remote subsystem. See also “allied thread” on page 3.

| **database administrator (DBA).** (1) A person who is responsible for the design, development, operation, security,
7 maintenance, and use of a database. (2) A DB2 UDB user with DBADM authority.

| **database agent.** A representation for the physical process or thread that will do the actual work inside the database
| engine.

Database Application Remote Interface (DARI). Obsolete term for “stored procedure” on page 80.

7 **database authority.** An authority that authorizes the possessor to perform database-level tasks, such as connecting
7 to the database or creating packages in the database.

database catalog. In the Data Warehouse Center, a collection of tables that contains descriptions of database objects
such as tables, views, and indexes.

database client. A workstation that is used to access a database that is on a database server.

| **database configuration parameter.** A parameter whose value limits the system resources that a database can use.
See also “configuration parameter” on page 15 and “database manager configuration parameter” on page 21.

| **database connection services (DCS) directory.** A directory that contains entries for remote host databases and the
| corresponding application requester used to access them.

database descriptor (DBD). An internal representation of a DB2 Universal Database for z/OS and OS/390 database
definition, which reflects the data definition that is in the DB2 Universal Database for z/OS and OS/390 catalog. The
objects that are defined in a database descriptor are table spaces, tables, indexes, index spaces, and relationships.

database directory. A directory that contains database access information for all databases to which a client can
connect. See also “node directory” on page 55.

| **database engine.** The part of the database manager that provides the base functions and configuration files that are
| needed to use the database.

| **database function.** The relationship between a set of input data and a set of result values. See also “built-in
function” on page 9 and “user-defined function” on page 91.

database log. A set of primary and secondary log files that consist of log records that record all changes to a
database. The database log is used to roll back changes for units of work that are not committed and to recover a
database to a consistent state.

database-managed space (DMS) table space. A table space whose space is managed by the database. See also
“system-managed space table space ” on page 83.

database management system (DBMS). Synonym for “database manager.”

7 **database manager.** A program that manages data by providing the services of centralized control, data
7 independence, and complex physical structures for efficient access, integrity, recovery, concurrency control, privacy,
7 and security.

| **database manager configuration parameter.** A configuration parameter that is established when the instance is created. Most database manager configuration parameters affect the amount of system resources that will be allocated to a single instance of the database manager, or they configure the setup of the database manager and the different communications subsystems based on environmental considerations. See also “configuration parameter” on page 15 and “database configuration parameter” on page 20.

| **database manager instance.** (1) A logical database manager environment similar to an image of the actual database manager environment. It is possible to have several instances of the database manager product on the same server. Use these instances to separate the development environment from the production environment, tune the database manager to a particular environment and protect sensitive information. (2) The DB2 code that manages data. An instance has its own databases (which other instances cannot access), and all its database partitions share the same system directories. It also has separate security from other instances on the same computer.

| **database name.** The identifying name that a user provides as part of the CREATE DATABASE command or application programming interface. A database name must be unique within the location in which it is cataloged.

database node. See “database partition.”

7 **database object.** (1) One of many objects that comprise an installation of DB2 Universal Database, including the instance and the databases, database partition groups, buffer pools, tables, and indexes within the instances. (2) An object that a user creates in the database, such as a procedure, trigger, or any other object that can be created by issuing a CREATE statement.

| **database object hierarchy.** An arrangement of database objects into parent/child relationships. For example, a database is the child of its database instance parent.

7 **database partition.** In a partitioned database environment, a part of the database that consists of its own user data, indexes, configuration file, and transaction logs.

| **database partition group.** In a partitioned database environment, a named set of one or more database partitions. This term replaces the term nodegroup.

| **database partition server.** In a partitioned database environment, an occurrence of DB2 that is recorded in the db2nodes.cfg file.

7 **database recovery log.** A set of primary and secondary log files that are used in replication to record all changes to a database in log records.

database request module (DBRM). A data set member that is created by the DB2 Universal Database for z/OS and OS/390 precompiler and that contains information about SQL statements. DBRMs are used in the bind process.

7 **database server.** In a client-server environment, a stand-alone workstation in a local area network on which the database manager is installed, allowing client workstations that run applications to access the database remotely. In the DB2 Universal Database environment, the database server function is provided by the distributed data facility to access DB2 Universal Database data from local applications or a remote database server that is acting as an intermediate database server.

| **database system monitor.** A collection of APIs that collect information regarding the state of the database system at the instance, database, and application levels. This information is stored in data elements, which can be examined by taking point-in-time snapshots, or by using the event monitor to log system activity over a period of time.

7 **data blocking.** In SQL replication, the process of replicating a specific number of minutes’ worth of change data during an Apply cycle.

| **data consolidation.** A replication configuration that contains one read-only target database. The target table contains rows of data from one or more source databases.

7 **data currency.** In DB2 Universal Database for z/OS and OS/390, the state in which data that is retrieved into a host variable in a program is a copy of data in the base table.

| **data definition language (DDL).** A language for describing data and its relationships in a database.

data definition name (ddname). In DB2 Universal Database for z/OS and OS/390, the name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

Glossary

data description language. Synonym for “data definition language” on page 21.

| **data dictionary.** A repository of information about an organization’s application programs, databases, logical data
| models, users, and authorizations. A data dictionary can be manual or automated.

7 | **data distribution replication.** In replication, a configuration that contains a single source table, from which changes
7 | are replicated to one or more read-only target tables. Before replication to the target tables can occur, the tables must
7 | contain a complete set of data from the source table.

| **data element.** See “monitor element” on page 53.

| **data interchange.** The sharing of data between applications. XML supports data interchange without needing to go
| through the process of first transforming data from a proprietary format.

7 | **DataJoiner.** See “DB2 Information Integrator” on page 25. See also “federated server” on page 35.

7 | **DataJoiner Replication Administration (DJRA) tool.** See “DB2 Information Integrator” on page 25. See also
7 | “Control Center” on page 16.

| **DATALINK.** An SQL data type that enables logical references from the database to a file stored outside the
| database.

data link control (DLC). In SNA, the protocol layer that consists of the link stations that schedule data transfer over
a link between two nodes and perform error control for the link.

| **Data Link Reconcile Not Possible (DRNP).** The state of a DB2 table in which one or more DATALINK type
| columns contain file references whose integrity is violated (for example, as the result of restoring a database without
| the ability to restore the files that the database refers to).

| **Data Link Reconcile Pending (DRP).** The state of a DB2 table in which one or more DATALINK type columns
| contain file references whose integrity might be in doubt (for example, as the result of restoring a database without
| rolling forward through the database logs).

| **Data Links Filesystem Filter (DLFF).** A DB2 Data Links Manager component. A file system filter program that
| enforces data integrity by ensuring valid and controlled access to linked files. See also “linked file” on page 48.

| **Data Links File Manager (DLFM).** A component of the DB2 Data Links Manager that enables a DB2 database to
| manage files that are outside of the database.

| **Data Links File System (DLFS).** A file system that is under the control of the Data Links Filesystem Filter (DLFF).

| **Data Links Manager Administrator.** The person and the user ID that is responsible for administering the DB2 Data
| Links Manager and its associated environment. Sometimes also referred to as *DLFM User*, because when DB2 Data
| Links Manager is installed, an account with the default user ID of *dlfm* is set up for use by the Data Links Manager
| Administrator.

| The Data Links Manager Administrator user ID also owns all of the resources used by the DLFM component, for
| example:

- | • The DB2 instance containing the DLFM_DB database
- | • Linked files referenced in a READ PERMISSION DB DATALINK type column
- | • The user ID under which the DLFM Server itself runs

| See also “dlmadmin account” on page 29 and “superuser” on page 82.

| **Data Links server.** A computer that contains these DB2 Data Links Manager components: a Data Links File Manager
| (DLFM), a Data Links Filesystem Filter (DLFF) controlling a Data Links File System (DLFS), and a DB2 database
| (used as the Logging Manager).

| **data manipulation language (DML).** A subset of SQL statements that is used to manipulate data. Most applications
| primarily use DML SQL statements, which are supported by the DB2 Connect program. SELECT, INSERT, UPDATE,
| and DELETE statements are similar across the IBM relational database products. See also “Structured Query
| Language” on page 81 and “data definition language” on page 21.

data mart. A subset of a data warehouse that contains data that is tailored and optimized for the specific reporting needs of a department or team. A data mart can be a subset of a warehouse for an entire organization, such as data that is contained in online analytical processing (OLAP) tools.

data message. In Q replication and event publishing, a message that contains all or part of a committed transaction that involve source tables, a committed operation on a single row in a source table (event publishing only), or all or part of a large object (LOB) value from a row operation within a transaction.

data mining. The process of collecting critical business information from a data warehouse, correlating the information and uncovering associations, patterns, and trends.

data partition. In a z/OS or OS/390 environment, a VSAM data set that is contained within a partitioned table space.

data-partitioned secondary index (DPSI). A secondary index that is partitioned. The index is partitioned according to the underlying data.

data sharing. The ability of two or more DB2 Universal Database for z/OS and OS/390 subsystems to directly access and change a single set of data.

data sharing group. A collection of one or more DB2 Universal Database for z/OS and OS/390 subsystems that directly access and change the same data while maintaining data integrity.

data sharing member. (1) A local or remote relational or nonrelational data manager that is capable of supporting data access using an ODBC driver that supports the ODBC APIs. (2) In a federated system, typically a relational DBMS instance and one or more databases that are supported by that instance. A federated system can include other types of data sources, such as flat-file databases and table-structured files.

data source. A repository of data to which a federated server can connect and then retrieve data by using wrappers. A data source can contain relational databases, XML files, search algorithms, table-structured files, or other objects. In a federated system, data sources appear as a single collective database.

data source object. In a federated system, an object at the data source on which you can perform operations. Examples include a database table, a database view, or a spreadsheet list. See also “nickname” on page 55.

data space. In releases prior to DB2 Universal Database for z/OS and OS/390 Version 8, space ranging in size from 0 bytes to 2 gigabytes of contiguous virtual storage addresses that a program can directly manipulate. Unlike an address space, a data space can hold only data; it does not contain common areas, system data, or programs.

data type. In SQL, an attribute of columns, literals, host variables, special registers, and the results of functions and expressions.

data type mapping. In a federated system, the mapping of a data type used at a data source to a DB2 data type. For example, the Oracle type FLOAT maps by default to the DB2 type DOUBLE. DB2 supplies default mappings for most kinds of data types; the default mappings are in the wrappers.

data warehouse. (1) A subject-oriented nonvolatile collection of data that is used to support strategic decision making. The warehouse is the central point of data integration for business intelligence. It is the source of data for data marts within an enterprise and delivers a common view of enterprise data. (2) A central repository for all or significant parts of the data that an organization’s business systems collect. Also known as an *information warehouse*. See also “data mart.”

Data Warehouse Center. The component of DB2 Universal Database that provides the graphical interface and the software behind it that enables you to work with the components of the warehouse. You can use the Data Warehouse Center to define and manage the warehouse data and the processes that create the data in the warehouse

Data Warehouse Center administrative interface. The user interface to the administration functions of the Data Warehouse Center. The interface can be on the Data Warehouse Center server or on different computers for multiple administrators.

Data Warehouse Center program. A program, supplied with the Data Warehouse Center, that can be started from the Data Warehouse Center and that is automatically defined. For example, DB2 Load programs and transformers are Data Warehouse Center programs.

Glossary

Data Warehouse Center property. An attribute that applies across sessions of the Data Warehouse Center, such as the tools catalog that contains the technical metadata. See also “property” on page 64.

| **date.** A three-part value that designates a day, month, and year. For example, YYYY-MM-DD.

date duration. A DECIMAL (8,0) value that represents a number of years, months, and days.

datetime value. A value of the data type DATE, TIME, or TIMESTAMP.

DBA Utility. A tool that lets DB2 users configure databases and database manager instances, manage the directories necessary for accessing local and remote databases, back up and recover databases or table spaces, and manage media on a system using a graphical interface. The tasks provided by this tool can be accessed from the DB2 Control Center.

DBA. See “database administrator” on page 20.

DBCLOB. See “double-byte character large object” on page 30.

DBCS. See “double-byte character set” on page 30.

DBD. See “database descriptor” on page 20.

| **DBID.** In DB2 Universal Database for z/OS and OS/390, a database identifier.

DBMS. See “database management system” on page 20.

DBMS instance connection. A logical connection between an application and an agent process or thread that is owned by a DB2 instance.

DBRM. See “database request module” on page 21.

7 **DB2 administration server.** A control point that is used to assist with administration tasks on DB2 servers and to perform remote tasks on the server and the host system on behalf of a client request.

| **DB2 Application Development Client (DB2 AD Client).** A collection of tools that help developers create database applications.

7 **DB2 Call Level Interface (CLI).** See “call level interface” on page 9.

7 **DB2 client.** A client that determines the location of a remote database, manages the transmission of requests to the database server, and returns the results.

DB2 command. An instruction to the operating system to access and maintain the database manager. For example, DB2 commands allow a user to start or stop a database, display information on current users and the status of databases.

7 **DB2 Connect.** A product that enables client applications to read and update data that is stored on DB2 family database servers.

7 **DB2 Data Links Manager.** A separately orderable feature that enables applications to manipulate data that is stored in unstructured files and in the relational database management system (RDBMS). DB2 Data Links Manager enables DB2 Universal Database to manage unstructured files as if they are stored in the database and provides the integration between the RDBMS and the external file systems through extensions to DB2 Universal Database.

| **DB2 DataPropagator.** A product that provides DB2 replication for OS/390, z/OS, OS/400, z/VM, VM, and VSE operating-system environments. For UNIX and Windows operating-system environments, replication is integrated with DB2 and does not require a separate license. See also “replication” on page 70.

| **DB2DC.** See “Development Center” on page 27.

| **DB2 Download Tool.** A tool that performs high-speed data transfers between an MVS and an SP system.

| **DB2DT.** See “DB2 Download Tool.”

- 7 **DB2 extender.** A program that stores and retrieves data types other than the traditional numeric and character data,
7 such as image, audio, and video data, and complex documents.
- 7 **DB2 Geodetic Extender.** A DB2 UDB component that stores and manipulates spatial data using the round-Earth
7 model that is a continuous, closed globe (unlike DB2 Spatial Extender, which treats the Earth as a flat map).
- | **DB2 host.** In a DB2 Data Links Manager configuration, a DB2 database, on a DB2 server, that contains a DATALINK
| column.
- 7 **DB2I.** In DB2 Universal Database for z/OS and OS/390, DB2 Interactive.
- | **DB2 Information Integrator.** An IBM product that integrates diverse, distributed, and real-time data and that
| provides wrappers for accessing and integrating structured and unstructured data under a single API. DB2
| Information Integrator replaces the DB2 Relational Connect, DB2 Life Sciences Data Connect, and DB2 DataJoiner
| products and tools.
- DB2I Kanji Feature.** In DB2 Universal Database for z/OS and OS/390, the tape that contains the panels and jobs
that allow a site to display DB2I panels in Kanji.
- 7 **DB2 .NET Data Provider.** An extension of the ADO.NET interface that allows .NET applications to access a DB2
7 UDB database through a secure connection, run commands, and retrieve results.
- | **DB2 Net Search Extender.** A program that provides full-text retrieval through a DB2 stored procedure. The DB2 Net
Search Extender is primarily optimized for performance. Using DB2 Net Search Extender can be particularly
advantageous in applications where search performance on large indexes and scalability according to concurrent
queries are important factors.
- DB2 Net Search Extender also provides powerful search features that are enhanced by additional rich linguistic
functionality for applications with highly structured documents where the information need is complex, and the
7 quality and precision of the search result are key issues over and above system response times.
- | **DB2 PM.** DB2 Performance Monitor for z/OS and OS/390.
- 7 **DB2 replication.** See “SQL replication” on page 79. See also “Q replication” on page 65.
- DB2 SDK.** See “DB2 Application Development Client” on page 24.
- 7 **DB2 Spatial Extender.** A DB2 UDB component that stores and manipulates spatial data. You use the DB2 Spatial
Extender to generate and analyze spatial information about geographic features. See also “geographic feature” on
| page 38.
- | **DB2 Text Extender.** Renamed and enhanced in DB2 Universal Database Version 8. See “DB2 Net Search Extender.”
- | **DB2 tools catalog.** A set of tables or files that is maintained by the database tools (Data Warehouse Center, Control
| Center, Task Center, Information Catalog Center) and contains information about the processes and tasks that DB2
| runs, such as loads, reorgs, database maintenance processes, data movement processes, and the associated schedules,
| logs, and dependencies.
- | **DB2 tools metadata.** The information about the processes and tasks that DB2 runs, such as loads, reorgs, database
| maintenance processes, data movement processes, and the associated schedules, logs, and dependencies. The DB2
| tools metadata is contained in the DB2 tools catalog.
- | **DB2 XML Extender.** A program that is used to store and manage XML documents in DB2 tables. Well-formed and
validated XML documents can be generated from existing relational data, stored as column data, and the content of
XML elements and attributes can be stored in DB2 tables.
- DCLGEN.** See “declarations generator” on page 26.
- DDF.** See “distributed data facility” on page 28.
- DDL.** See “data definition language” on page 21.
- ddname.** See “data definition name” on page 21.

Glossary

| **deadlock.** A condition under which a transaction cannot proceed because it is dependent on exclusive resources that
| are locked by another transaction, which in turn is dependent on exclusive resources that are in use by the original
| transaction.

deadlock detector. A process within the database manager that monitors the states of the locks to determine if a
deadlock condition exists. When a deadlock condition is detected, the detector stops one of the transactions involved
in the deadlock. This transaction is rolled back and the other transaction can proceed.

7 **decision support system.** In the Information Catalog Center, a system of applications that helps users make
7 decisions by analyzing business data that is presented in meaningful ways; for example, spreadsheets, charts, and
7 reports.

declarations generator (DCLGEN). A subcomponent of DB2 Universal Database for z/OS and OS/390 that
generates SQL table declarations and COBOL, C, or PL/I data structure declarations that conform to the table. The
declarations are generated from DB2 Universal Database for z/OS and OS/390 system catalog information. DCLGEN
is also a DSN subcommand.

| **declared temporary table.** A table that holds temporary data and is defined with the SQL statement DECLARE
| GLOBAL TEMPORARY TABLE. Information about declared temporary tables is not stored in the DB2 catalog, so this
| kind of table is not persistent and can be only used by the application process that issued the DECLARE statement.
| See also “base table” on page 7, “created temporary table” on page 19, and “temporary table” on page 85.

| **decompose.** In XML Extender, to separate XML documents into a collection of relational tables in an XML collection.

| **default subsystem name (DSN).** (1) In the z/OS or OS/390 environment, the name of the TSO command processor
| of DB2. (2) The name of the DB2 subsystem that can connect to the control server (the default subsystem name is
| DSN). (3) In the z/OS or OS/390 environment, the first three characters of DB2 module and macro names.

| **default view.** In XML Extender, a representation of data in which an XML table and all of its related side tables are
| joined.

deferred embedded SQL. SQL statements that are neither fully static nor fully dynamic. Like static statements, they
are embedded within an application, but like dynamic statements, they are prepared during the execution of the
application.

| **deferred write.** In DB2 Universal Database for z/OS and OS/390, the process of asynchronously writing changed
data pages to disk.

definition metadata. In the Data Warehouse Center, information about the format of the data warehouse (the
schema), the sources of the data, and the transformations applied in loading the data.

| **degree of parallelism.** The number of concurrently executed operations that are initiated to process a query.

delete-connected. In SQL, a property of table that is a dependent of table P or a dependent of a table to which
delete operations from table P cascade.

| **delete history.** In the Information Catalog Center, a log of delete activity, the capture of which is turned on and off
| by the Information Catalog Center administrator. The log can be transferred to a tag language file.

| **delete hole.** A row for a SELECT statement of a cursor that no longer has a corresponding row in the base table
| because the row was deleted. A delete hole is created when a row in the base table is deleted while a cursor is open
| whose SELECT statement result contains the row that is deleted. Such a row is no longer accessible though the
| cursor. See also “hole” on page 40 and “update hole” on page 91.

delete rule. A rule that is associated with a referential constraint that either restricts the deletion of a parent row or
specifies the effect of such a deletion on the dependent rows.

7 **delete trigger.** A trigger that is activated when a record is deleted. See also “trigger” on page 88.

7 **delimited identifier.** A sequence of characters enclosed by quotation marks ("). The sequence must consist of one or
7 more characters of SQL. Leading blanks in the sequence are significant. Trailing blanks in the sequence are not
7 significant. The length of a delimited identifier does not include the two quotation marks. See also “ordinary
7 identifier” on page 58.

delimiter. A character or flag that groups or separates items of data.

delimiter token. A string constant, a delimited identifier, an operator symbol, or any of the special characters shown in syntax diagrams.

| **delta backup.** A copy of all database data that has changed since the last successful backup (full, incremental, or delta) of the table space in question. A delta backup is also known as a differential, or noncumulative, backup image. The predecessor of a delta backup image is the most recent successful backup that contains a copy of each of the table spaces in the delta backup image.

| **denormalization.** The intentional duplication of columns in multiple tables whose consequence is increased data redundancy. Denormalization is sometimes necessary to minimize performance problems and is a key step in designing a physical relational database design. See also “normalization” on page 56.

dependent. In SQL, an object (row, table, or table space) that has at least one parent. See also “parent row” on page 60, “parent table” on page 60, and “parent table space” on page 60.

| **dependent foreign key table.** A dependent foreign key table of a given table is a table that has at least one foreign key constraint referencing the given table.

| **dependent immediate materialized query table.** A table whose definition is based on the result of a query and whose data is in the form of precomputed results. These results come from the tables or nicknames that are used in the definition of the materialized query table.

dependent logical unit (DLU). A logical unit that requires assistance from a system services control point (SSCP) to instantiate an LU-to-LU session. See “independent logical unit” on page 42.

| **dependent row.** A row that contains a foreign key that matches the value of a parent key in the parent row. The foreign key value represents a reference from the dependent row to the parent row. See also “parent row” on page 60.

| **dependent materialized query table.** A materialized query table that references a given table directly or indirectly (For example, from a view) in its materialized query table definition.

| **dependent table.** A table that is a dependent in at least one referential constraint.

| **dependent table space.** A table space that contains a dependent of a parent table. See also “parent table space” on page 60.

| **derived data.** In the Information Catalog Center, data that is copied or enhanced (perhaps by summarizing the data) from operational data sources into an informational database.

descendent. An object that has a dependent of an object or is the dependent of a descendent of an object.

| **descendent immediate materialized query table.** A descendent immediate materialized query table is a materialized query table defined with the REFRESH IMMEDIATE option that directly refers to a descendent immediate materialized query table in its materialized query table definition.

| **descendent materialized query table.** A materialized query table that references in its materialized query table definition a descendent materialized query table directly or indirectly.

descendent row. A row that is dependent on another row, or a row that is a descendent of a dependent row.

7 **descendent table.** A table that is in a dependent relationship to a parent table, or to another descendent table. See
7 “dependent table.”

| **descriptive data.** See “metadata” on page 52.

| **deterministic function.** A user-defined function whose result is solely dependent on the values of the input arguments. Successive invocations with the same argument values always produce the same results. Contrast with “not deterministic function” on page 56.

| **Development Center.** A component of DB2 UDB that provides a graphical interface for building, testing, and deploying stored procedures and user-defined functions. Features include a server view, an integrated SQL debugger, export and import wizards, and a text editor.

Glossary

- | **Development Center project.** A project that contains information about database connections and routines that are being developed with the Development Center.
- | **device name.** A name reserved by the system or a device driver that refers to a specific device. For example, the DOS device name for the parallel port is LPT1.
- 7 **DFP.** In a z/OS or OS/390 environment, Data Facility Product.
- 7 **dictionary.** A collection of language-related linguistic information that the DB2 Net Search Extender uses during text analysis, indexing, retrieval, and highlighting of documents in a particular language.
- | **dictionary relationship type.** In the Information Catalog Center, the relationship type that is used to associate a glossary entry object type with another object. A glossary entry object type can be used to define terminology that is associated with the object. See also “relationship type” on page 69.
- differential backup image.** See “delta backup” on page 27.
- | **differential refresh.** See “change-capture replication” on page 11.
- | **dimension.** A data category, such as time, accounts, products, or markets. The elements of a dimension are referred to as members. Dimensions offer a very concise, simple way of organizing and selecting data for retrieval, exploration, and analysis. Dimensions also represent the highest consolidation level in a multidimensional database outline. See also “business dimension” on page 9, “multidimensional analysis” on page 54, and “dimension table.”
- | **dimension block index.** In multidimensional clustering, a block index that is automatically created for a particular dimension when the dimension is defined on an MDC table. This index is used to maintain the clustering of data along that dimension, together with the other dimensions defined on the table.
- | **dimension table.** The representation of a dimension in a star schema. Each row in a dimension table represents all of the attributes for a particular member of the dimension. See also “dimension” and “star schema” on page 80.
- | **directed join.** A relational operation in which all of the rows in one or both of the joined tables are rehashed and directed to new database partitions based on the join predicate. If all of the partitioning key columns in one table participate in the equijoin predicates, the other table is rehashed; otherwise (if there is at least one equijoin predicate), both tables are rehashed. See “join” on page 46.
- directory.** The DB2 Universal Database for z/OS and OS/390 system database that contains internal objects such as database descriptors and skeleton cursor tables.
- directory services.** A portion of the APPN protocols that maintains information about the location of resources in an APPN network.
- 7 **disable.** To restore a database, a text table, or a text column to its condition before it was enabled for the DB2 Net Search Extender by removing the items that were created during the enabling process.
- 7 **disaster recovery.** The process of restoring a database after a partial or complete site failure that was caused by a catastrophic event such as an earthquake or fire. Typically, disaster recovery requires a full database backup at another location.
- distinct type.** A user-defined data type that is internally represented as an existing type (its source type), but is considered to be a separate and incompatible type for semantic purposes.
- distributed data facility (DDF).** A set of DB2 Universal Database for z/OS and OS/390 components through which DB2 Universal Database for z/OS and OS/390 communicates with another RDBMS.
- | **distributed directory database.** The complete listing of all the resources in the network as maintained in the individual directories scattered throughout an APPN network. Each node has a piece of the complete directory, but it is not necessary for any one node to have the entire list. Entries are created, modified, and deleted through system definition, operator action, automatic registration, and ongoing network search procedures. Synonym for distributed network directory.

- | **distributed installation.** A process by which DB2 products can be installed using systems management software, such as Microsoft Systems Management Server (SMS) on Windows NT or Windows 2000, or simply with a shared CD drive or shared network hard drive using response files. Also known as a silent installation or unattended installation.

- | **distributed network directory.** See “distributed directory database” on page 28.

- | **distributed relational database.** A database whose tables are stored on different but interconnected computing systems.

- | **Distributed Relational Database Architecture (DRDA).** The architecture that defines formats and protocols for providing transparent access to remote data. DRDA defines two types of functions, the application requester function and the application server function.

- | **distributed request.** In a federated database system, an SQL query directed to two or more data sources.

- | **distributed transaction.** A transaction that updates data in more than one database. See also “two-phase commit” on page 89.

- | **distributed unit of work.** A unit of work that allows SQL statements to be submitted to multiple relational database management systems, but no more than one system per SQL statement.

- | **DLC.** See “data link control” on page 22.

- | **DLFF.** See “Data Links Filesystem Filter (DLFF)” on page 22.

- | **DLFM.** See “Data Links File Manager (DLFM)” on page 22.

- | **DLFM_ASNCOPYD file-copy daemon (Data Links Manager Replication daemon).** The DLFM process which enables replication of DB2 Data Links Manager files (in conjunction with the associated DB2 relational data) in support of data replication.

- | **DLFM_DB database.** A DB2 database that acts as a logging manager for the Data Links server.

- | **DLFS.** See “Data Links File System (DLFS)” on page 22.

- | **DLM.** See “DB2 Data Links Manager” on page 24.

- | **dlnadmin account.** In DB2 Data Links Manager, an account that has advanced user privileges in Windows environments, and is intended to be equivalent to the root user in UNIX environments. Its purpose is to act as a superuser to perform any necessary advanced administration operations by both the DLFM component and the Data Links Manager Administrator on the Data Links server.

- | **DLU.** See “dependent logical unit” on page 27.

- | **DML.** See “data manipulation language” on page 22.

- | **DMS table space.** See “database-managed space table space” on page 20.

- | **DNS.** See “domain name server (DNS)” on page 30.

- 7 **document access definition (DAD).** An XML document format that defines the mapping between XML and relational data.

- 7 **Document Access Definition Extension (DADX).** A configuration file that controls both XML-based and SQL-based forms of querying by defining the operations that can be performed by a Web service.

- 7 **document model.** The definition of the structure of a document in terms of the sections that it contains. The DB2 Net Search Extender uses a document model when indexing.

- 7 **document type definition (DTD).** The rules that specify the structure for a particular class of SGML or XML documents. The DTD defines the structure with elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation can be used within the particular class of documents.

- | **domain.** A part of a network that is administered as a unit with a common protocol.

Glossary

7 **domain name.** In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of
7 subnames that are separated by a delimiter character, for example, www.ibm.com.

7 **domain name server (DNS).** A server program that converts names to addresses by mapping domain names to IP
7 addresses.

Domino™ Go Web server. The Web server that offers both regular and secure connections. ICAPI and GWAPI are the interfaces provided with this server.

7 **double-byte character large object (DBCLOB).** A data type that contains a sequence of double-byte characters that
7 can range in size from 0 bytes to 2 gigabytes less one byte. This data type can be used to store large double-byte text
7 objects. Also called double-byte character large object string. See also “character large object” on page 11 and “binary
7 large object” on page 7.

l **double-byte character set (DBCS).** A set of characters in which each character is represented by two bytes. These
l character sets are commonly used by national languages, such as Japanese and Chinese, that have more symbols than
l can be represented by a single byte. See also “single-byte character set” on page 77 and “multibyte character set” on
l page 54.

double-precision floating point number. In SQL, a 64-bit approximate representation of a real number.

7 **DPSI.** See “data-partitioned secondary index ” on page 23.

l **drain.** In DB2 Universal Database for z/OS and OS/390, the act of acquiring a locked resource by quiescing access
l to that object. See also “claim” on page 12.

drain lock. In DB2 Universal Database for z/OS and OS/390, a lock on a claim class that prevents a claim from occurring.

DRDA. See “Distributed Relational Database Architecture” on page 29.

l **DRDA access.** An open method of accessing distributed data by which you can connect to another database server
l (by location), using an SQL statement, to execute packages that have been previously bound at that location. The SQL
l CONNECT statement or a three-part name SQL statement is used to identify the server. See also “private protocol
l access” on page 63.

l **DRNP.** See “Data Link Reconcile Not Possible (DRNP)” on page 22.

l **DRP.** See “Data Link Reconcile Pending (DRP)” on page 22.

DSN. See “default subsystem name” on page 26.

l **DTD.** See “document type definition” on page 29.

l **DTD reference table.** A table that consists of DTDs, which are used to validate XML documents and to help
l applications to define a DAD. This table is created when a database is enabled for XML. Users can insert their own
l DTDs into the DTD_REF table.

DUOW. See “distributed unit of work” on page 29.

l **dual log path.** A secondary log path that is used to maintain duplicate copies of online archived files and the active
l log.

duration. In SQL, a number that represents an interval of time. See “date duration” on page 24, “labeled duration”
on page 47, and “time duration” on page 86.

dynamic bind. A process by which SQL statements are bound as they are entered. See “bind” on page 8. See also
“static bind” on page 80.

7 **dynamic cursor.** A named control structure that an application program uses to change the size of the result table
7 and the order of its rows after the cursor is opened. See also “static cursor” on page 80.

l **dynamic SQL.** SQL statements that are prepared and executed at run time. In dynamic SQL, the SQL statement is
l contained as a character string in a host variable and is not precompiled. See also “embedded SQL” on page 31 and
l “static SQL” on page 80.

- 7 **dynamic statement cache pool.** A cache, located above the 2-GB storage line, that holds dynamic SQL statements.

E

EA-enabled table space. In DB2 Universal Database for z/OS and OS/390, a table space or index space that is enabled for extended addressability and that contains individual partitions (or pieces, for LOB table spaces) that are greater than 4 gigabytes.

- 1 **EBCDIC.** A coded character set of 256 8-bit characters developed for the representation of textual data, typically used on zSeries and iSeries servers. See also “ASCII” on page 5 and “Unicode” on page 89.

- 1 **edition.** See “step edition” on page 80.

EDM. Electronic data management.

- 7 **EDU.** See “engine dispatchable unit.”

EID. Event identifier.

- 2 **elapsed queued time.** In Query Patroller, the length of time that elapses between the time a query is created and the time a query starts to run. See also “elapsed total time.”

- 2 **elapsed total time.** In Query Patroller, the combined total of the elapsed queued time and the elapsed run time for a query. See also “elapsed queued time” and “execution time” on page 32.

- 2 **electronic data management pool (EDM pool).** In DB2 Universal Database for z/OS and OS/390, a pool of main storage that is used for database descriptors, application plans, authorization cache, application packages, and dynamic statement caching.

- 1 **element.** See “XML element” on page 94.

- 7 **embedded SQL.** SQL statements that are coded in an application program. See “static SQL” on page 80.

EN. See “end node.”

- 7 **enable.** (1) To prepare a database, a text table, or a text column for use by the DB2 Net Search Extender or the DB2 XML Extender. (2) To turn on or activate.

enclave. In Language Environment (which is used by DB2 Universal Database for z/OS and OS/390), an independent collection of routines, one of which is designated as the main routine. An enclave is similar to a program or run unit.

encoding scheme. A set of rules to represent character data.

- 7 **encryption.** The conversion of data into a cipher. A key is required to encrypt and decrypt the data. Encryption provides protection from persons or software that attempt to access the data without the key.

end node. In APPN, a node that supports sessions between its local control point and the control point in an adjacent network node.

- 1 **end-to-end latency.** In replication, an approximate measurement of the time that replication requires to capture changes from a source database and apply those changes to a target database. See also “Apply latency” on page 4, “Capture latency” on page 10, “Q Apply latency” on page 65, and “Q Capture latency” on page 65.

- 7 **engine dispatchable unit (EDU).** Coordinates application requests to a DB2 database. Referred to as a process on UNIX operating systems and a thread on Windows operating systems.

- 7 **enhanced conflict detection.** In SQL replication, conflict detection that guarantees data integrity among all replicas and the source table. The Apply program locks all replicas or user tables in the subscription set against further transactions. It begins detection after all changes made prior to locking have been captured. See “conflict detection” on page 15.

Glossary

- | **entity.** (1) A person, object, or concept about which information is stored. In a relational database, entities are represented as tables. A database includes information about the entities in an organization or business, and their relationships to each other. (2) A unit of data that can be classified and have stated relationships to other entities within that database.
- | **enumerated list.** In DB2 Universal Database for z/OS and OS/390, a set of DB2 objects that are defined with a LISTDEF utility control statement in which pattern-matching characters (*, %, _ , or ?) are used.
- environment handle.** A handle that identifies the global context for database access. All data that is pertinent to all objects in the environment is associated with this handle.
- 7 **environment profile.** A script that is provided with the DB2 Net Search Extender that contains settings for
7 environment variables.
- EOM.** End of memory.
- EOT.** End of task.
- | **equijoin.** A join operation in which the join-condition has the form *expression = expression*.
- error page range.** A range of pages that are considered to be physically damaged. DB2 Universal Database for z/OS and OS/390 does not allow users to access any pages that fall within this range.
- | **escape character.** See “SQL escape character” on page 78.
- 7 **ESDS.** A VSAM data set whose records are physically in the same order in which they were put in the data set. A
7 VSAM data set is processed by addressed direct access or addressed sequential access and has no index. New records
7 are added at the end of the data set.
- ESMT.** See “external subsystem module table” on page 34.
- EUC.** See “Extended UNIX Code (EUC) encoding scheme” on page 33.
- | **event analyzer.** A database object that provides information about the database events that have taken place. An event analyzer is used with the event monitor file to assess and record performance information.
- | **event monitor.** A database object for monitoring and collecting data on database activities over a period of time. For
| example, starting the database might be an event that causes an event monitor to track the number of users on the
| system by taking an hourly snapshot of authorization IDs using the database.
- 7 **event publishing.** A data publishing solution that captures transactional data from DB2 Universal Database
7 recovery logs and publishes that data as XML messages. The XML messages are published to WebSphere MQ queues
7 where one or more user applications can retrieve and use those messages.
- 7 **event timing.** In SQL replication, the most precise method of controlling when to start a replication subscription
7 cycle. Contrast with “interval timing” on page 45.
- | **exception table.** (1) A user-created table that reflects the definition of the table being loaded. (2) A table that holds
| rows that violate referential constraints or check constraints that the CHECK DATA utility finds.
- | **exclusive lock.** A lock that prevents running executing application processes from accessing data. See also “shared
| lock” on page 76.
- executable statement.** An SQL statement that can be embedded in an application program, dynamically prepared and executed, or issued interactively.
- 2 **execution time.** The elapsed run time of a query. This is the time between the start and the end of the query
2 execution. See also “user time” on page 92 and “system time” on page 83.
- exit routine.** A program that receives control from another program to perform specific functions.
- | **expanding conversion.** A process that occurs when the length of the converted string is greater than that of the
| source string. See also “contracting conversion” on page 16.

explain. To capture detailed information about the access plan that was chosen by the SQL compiler to resolve an SQL statement. The information describes the decision criteria used to choose the access plan.

l **explain snapshot.** (1) A collection of information that is compressed when an SQL statement is explained. (2) A capture of the current internal representation of an SQL query and related information. This information is required by the Visual Explain tool.

7 **explain statistics.** The statistics in the catalog that are referenced when an SQL statement is explained.

explainable statement. An SQL statement for which the explain operation can be performed. Explainable statements are SELECT, UPDATE, INSERT, DELETE, and VALUES.

explained statement. An SQL statement for which an explain operation was performed.

7 **explicit connect.** A connection to a database in which both the user ID and password are specified.

7 **explicit hierarchical locking.** In DB2 Universal Database for z/OS and OS/390, locking that is used to make the
7 parent-child relationship between resources known to the internal resource lock manager. This type of locking avoids
7 global locking use when no inter-DB2 interest exists on a resource.

l **explicit privilege.** A privilege that has a name and is held as the result of SQL GRANT and REVOKE statements, for example, the SELECT privilege. See “privilege” on page 63. See also “implicit privilege” on page 41.

export. (1) To copy data from database tables to a file using formats such as PC/IXF, DEL, WSF, or ASC. See also “import” on page 41. (2) In the Information Catalog Center, to populate a tag language file with information catalog contents for use with another program.

exposed name. A correlation name, a table, or a view name that is specified in a FROM clause for which a correlation name is not specified.

expression. An SQL operand or a collection of operators and operands that yields a single value.

l **extended binary-coded decimal interchange code (EBCDIC).** See “EBCDIC” on page 31.

7 **extended recovery facility (XRF).** In a z/OS or OS/390 environment, a facility that minimizes the effect of failures
7 in MVS, VTAM, the host processor, or high-availability applications during sessions between high-availability
7 applications and designated terminals. This facility provides an alternative subsystem to take over sessions from the failing subsystem.

l **Extended UNIX Code (EUC) encoding scheme.** An encoding scheme that defines a set of encoding rules that can
l support one to four character sets. The encoding rules are based on the ISO2022 definition for the encoding of 7-bit
l and 8-bit data. The EUC encoding scheme uses control characters to identify some of the character sets.

l **Extensible Markup Language (XML).** A text-based tag language that is used for document processing and for
l publishing information on the Web.

7 **Extensible Stylesheet Language (XSL).** A language for specifying style sheets for XML documents. XSL consists of
7 two parts: a language for transforming XML documents and an XML vocabulary for specifying formatting semantics.
7 See also “Extensible Stylesheet Language Transformation.”

7 **Extensible Stylesheet Language Transformation (XSLT).** An XML processing language that is used to convert an
7 XML document into another document in XML, PDF, HTML, or other format.

extent. An allocation of space, within a container of a table space, to a single database object. This allocation consists of multiple pages.

extent map. A metadata structure that is stored within a table space that records the allocation of extents to each object in the table space.

external CCD table. In SQL replication, a CCD table that can be subscribed to directly because it is a registered replication source. It has its own row in the register table, where it is identified by the SOURCE_OWNER and SOURCE_TABLE columns. See “consistent-change-data table” on page 16. See also “internal CCD table” on page 45.

Glossary

- | **external function.** A function for which the body is written in a programming language that takes scalar argument values and produces a scalar result for each invocation. See also “sourced function” on page 77, “built-in function” on page 9, and “SQL function” on page 78.
- 7 **external method.** A method that has its functional logic implemented in an external host programming language application. The association of the method with the external code application is asserted by the specification of the EXTERNAL clause in the CREATE METHOD statement. See also “method” on page 53, “SQL method” on page 79, “external function,” “external procedure,” and “external routine.”
- | **external name.** The name of an executable file for a stored procedure or user-defined function that is written in a host programming language.
- | **external procedure.** A procedure that has its procedural logic implemented in an external host programming language application. The association of the procedure with the external code application is asserted by the specification of the EXTERNAL clause in the CREATE PROCEDURE statement. See also “procedure” on page 64, “external function,” and “SQL procedure” on page 79.
- 7 **external routine.** A function, method, or procedure that has its routine logic implemented in an external host programming language application. The association of the routine with the external code application is asserted by the specification of the EXTERNAL clause in the CREATE statement of the routine. See also “routine” on page 73, “SQL routine” on page 79, “external function,” and “external procedure.”
- 7 **external subsystem module table (ESMT).** In a z/OS or OS/390 environment, a table that specifies the name of the external subsystem module table, which specifies which attachment modules must be loaded by IMS.
- | **extract control file.** A file that contains statements that control the operation of an extractor utility program.
- | **extract program.** In the Information Catalog Center, a utility program that copies metadata from a metadata source (such as an *RDBMS catalog*), translates the metadata into tag language, and places this output into a tag language file.

F

- | **fact table.** (1) In DB2 OLAP Server, a table, or in many cases a set of tables, that contains all data values for a relational cube. (2) A relational table that contains facts, such as units sold or cost of goods, and foreign keys that link the fact table to each dimension table.
- 7 **failback.** In high availability disaster recovery (HADR), the process of restarting the original primary system and returning it to its status of primary system after a failover has occurred. See “failover.”
- failed member state.** In DB2 Universal Database for z/OS and OS/390, a state of a member of a data sharing group. When a member fails, the XCF permanently records the failed member state. This state usually means that the member’s task, address space, or MVS system terminated before the state changed from active to quiesced.
- 7 **failover.** The change in status of the standby system to primary system because a failure occurred on the original primary system.
- fallback.** (1) The process by which a database server, after failure causes it to run on another computer, returns automatically to run on the original computer when it becomes available. (2) The process of returning to a previous release of DB2 Universal Database for z/OS and OS/390 after attempting or completing migration to a current release.
- false global lock contention.** In DB2 Universal Database for z/OS and OS/390, an indication of contention from the coupling facility when multiple lock names are hashed to the same indicator and when no real contention exists.
- fast communication manager (FCM).** A group of functions that provide interpartition communication support in a partitioned database environment.
- | **FAT.** See “file allocation table” on page 35.
- | **federated database.** In a federated system, the database that is within the federated server. Users and applications interface with the federated database. To these clients, the data sources appear as a single collective database in DB2.

7 **federated savepoint.** An API at the data source that is used by a federated server to preserve the atomicity of SQL
7 statements. A federated server uses data source savepoint APIs to bracket a series of INSERT, UPDATE, and DELETE
7 statements executed at the data sources side on behalf of a single DB2 Universal Database INSERT, UPDATE, or
7 DELETE statement.

| **federated server.** The DB2 server in a federated system. Any number of DB2 instances can be configured to function
| as federated servers. You can use existing DB2 instances as your federated server, or you can create new ones
| specifically for the federated system.

| **federated system.** A special type of distributed database management system (DBMS). A federated system allows
| you to query and manipulate data located on other servers. The data can be in database managers such as Oracle,
| Sybase, Informix, and Microsoft SQL Server, or it can be in lists or stores such as a spreadsheet, Web site, or data
| mart.

| A federated system consists of a DB2 instance that will operate as a server, a database that will serve as the federated
| database, one or more data sources, and clients (users and applications) who will access the database and data
| sources.

7 **fenced.** Pertaining to a type, or characteristic, of a procedure, user-defined function, or federated wrapper that is
7 defined to run in a separate process from the database manager. When this type of object is run (using the fenced
7 clause), the database manager is protected from modifications by the object. See also “not fenced” on page 56.

| **fetch.** An SQL action that positions a cursor on the next row of its result table and assigns the values of that row to
| host variables.

| **fetch orientation.** The specification of the desired placement of the cursor as part of a FETCH statement (for
| example BEFORE, AFTER, NEXT, PRIOR, CURRENT, FIRST, LAST, ABSOLUTE, and RELATIVE). See also
| “scrollability” on page 74.

| **fetch sensitivity.** The specification that a FETCH statement has visibility to all changes made by this cursor, as well
| as changes made by other cursors, or other application processes. Fetch sensitivity results in always fetching the rows
| from the base table of the SELECT statement of the cursor.

field procedure. In DB2 Universal Database for z/OS and OS/390, a user-written exit routine that is designed to
| receive a single value and transform (encode or decode) it in any way that the user can specify.

| **file access token.** See “read token” on page 67.

7 **file allocation table (FAT).** A table that is used to allocate space on a disk for a file and to locate the file.

file reference variable. A host variable that is used to indicate that data resides in a file on the client rather than in
| a client memory buffer.

| **File System Migrator (FSM).** The virtual file system whose space usage is controlled by the Tivoli Space Manager.
| DB2 Data Links Manager supports the use of this file system in the AIX operating environment.

| **file update operations.** All actions that are involved when a file is changed, especially in the case where the file is
| referenced in a DATALINK type column and is under the control of a DB2 Data Links Manager. See also “linked file”
| on page 48.

| **filter factor.** In DB2 Universal Database for z/OS and OS/390, a number between zero and one that estimates the
| proportion of rows in a table for which a predicate is true. Those rows are said to qualify by that predicate. Filter
| factors affect the choice of access paths by estimating the number of rows qualified by a set of predicates.

fixed-length string. A character or graphic string whose length is specified and cannot be changed. See also
| “variable-length string” on page 92.

flagger. A precompiler option that identifies SQL statements in applications that do not conform to selected
| validation criteria (for example, the ISO/ANSI SQL92 entry-level standard).

7 **flush.** To transfer computer data from a temporary storage area to the computer’s permanent memory.

7 **foreign key.** (1) A column or set of columns that refers to a parent key. In a relational database, a key in one table
7 that references the primary key in another table. (2) In a federated system, a key in one nickname that references the

Glossary

7 primary key in another nickname and that the optimizer uses to improve query performance. This key is not
7 validated when operations such as insert and update are performed.

| **foreign server.** In a federated system, another term for data source that is used most often in the context of the
| SQL/MED standard. See also “data source” on page 23.

foreign update. An update that is applied to a target table and replicated to the local table.

forward log recovery. The third phase of restart processing during which DB2 Universal Database for z/OS and
OS/390 processes the log in a forward direction to apply all REDO log records.

forward-only cursor. See “nonscrollable cursor” on page 56.

7 **forward recovery.** See “rollforward recovery” on page 72.

| **fragmentation.** The separation of the index into pieces as a result of inserts and deletions in the index.

| **free space.** The total amount of unused space in a page. The space that is not used to store records or control
| information is free space.

| **free space control record (FSCR).** A record containing available space approximations for each of the next 500
| pages. In multidimensional clustering (MDC) tables, there is one FSCR for each block. It is stored on the first page of
| that block and covers only the pages in that block.

| **FSM.** See “File System Migrator” on page 35.

| **full outer join.** The result of an SQL join operation that includes the matched rows of both tables that are being
| joined and preserves the unmatched rows of both tables. See also “join” on page 46, “outer join” on page 58, “left
| outer join” on page 48, and “right outer join” on page 72.

7 **full refresh.** (1) In SQL replication, the process in which all of the data that matches registration and the
7 subscription-set predicates for a replication source table is copied to the target table. Also known as loading a target
7 table. A full refresh replaces all existing data in the target table. Contrast with “change-capture replication” on page
7 11. (2) In Q replication, the process in which all of the data that matches the search conditions for a Q subscription
7 for a replication source table is copied to the target table. A full refresh replaces all existing data in the target table.

| **fullselect.** A subselect, a values-clause, or a number of both that are combined by set operators. Fullselect specifies a
| result table. If UNION is not used, the result of the fullselect is the result of the specified subselect.

7 **fully escaped mapping.** A mapping from an SQL identifier to an XML name when the SQL identifier is a column
7 name.

fully qualified LU name. See “network-qualified name” on page 55.

7 **function.** A relationship between a set of input data values and a set of result values that is used to extend and
7 customize SQL. Functions are invoked from elements of SQL statements such as the select-list or the FROM clause.
7 See also “routine” on page 73, “column function” on page 13, “scalar function” on page 74, “table function” on page
7 84, and “row function” on page 73.

function body. The piece of code that implements a function.

| **function definer.** In DB2 Universal Database for z/OS and OS/390, the authorization identifier of the owner of the
| schema of the function that is specified in the CREATE FUNCTION statement.

7 **function directory.** A DB2 UDB directory that is used to store the executable files and libraries that are associated
7 with users’ external routines (procedures, functions, and methods).

function family. A set of functions with the same function name. The context determines whether the usage refers
to a set of functions within a particular schema, or all the relevant functions with the same name within the current
function path.

| **function implementer.** In DB2 Universal Database for z/OS and OS/390, the authorization identifier of the owner
| of the function program and function package.

function invocation. The use of a function with any argument values that are passed to the function body. The function is invoked by its name.

| **function mapping.** In a federated system, a mapping between a data source function and an existing DB2 Universal Database function. DB2 Universal Database supplies default mappings between existing built-in data source functions and built-in DB2 Universal Database functions; the default mappings are in the wrapper. The DB2 Universal Database counterpart function can be either a complete function or a function template. See “function template.”

| **function mapping options.** In a federated system, parameters of the CREATE FUNCTION MAPPING statement to which you can assign values that pertain to the mapping being created or to the data source function within the mapping. Such values, for example, can include estimated statistics on the overhead that will be consumed when the data source function is invoked. The query optimizer uses these estimates to decide if the function should be invoked by the data source or by DB2 Universal Database, when the data is returned from the data source. See “function mapping.”

function package. In DB2 Universal Database for z/OS and OS/390, a package that results from binding the DBRM for a function program.

| **function package owner.** In DB2 Universal Database for z/OS and OS/390, the authorization identifier of the user who binds the function program’s DBRM into a function package.

function path. An ordered list of schema names that restricts the search scope for unqualified function invocations and provides a final arbiter for the function selection process.

function path family. All the functions of the given name in all the schemas that are identified (or used by default) in the user’s function path.

| **function resolution.** The process, internal to the database manager, for which a particular function instance is selected for invocation. The function name, the data types of the arguments, and the function path are used to make the selection. Synonym for “function selection.”

function selection. See “function resolution.”

| **function shipping.** The process of sending the subsections of a request to the specific database partition that contains the applicable data.

function signature. The logical concatenation of a fully qualified function name with the data types of all of its parameters. Each function in a schema must have a unique signature.

7 **function template.** A DB2 UDB function that forces the federated server to start a data source function. A function template does not contain code that can be run.

G

7 **gap.** In SQL replication, a range of log or journal records that the Capture program cannot read. The change data in a gap can be lost.

| **gauge.** An indicator for the current value for an item. See also “counter” on page 18.

| **GBP.** See “group buffer pool” on page 38.

GBP-dependent. In DB2 Universal Database for z/OS and OS/390, the status of a page set or page set partition that is dependent on the group buffer pool. Either read/write interest is active among DB2 subsystems for this page set, or the page set has changed pages in the group buffer pool that are not yet cast out to disk.

7 **generalized trace facility (GTF).** In a z/OS or OS/390 environment, a service program that records significant system events such as I/O interrupts, SVC interrupts, program interrupts, or external interrupts.

| **generated column.** A column that is derived from an expression that involves one or more columns in the table.

7 **generic resource name.** In a z/OS or OS/390 environment, a name that VTAM uses to represent several application programs that provide the same function in order to handle session distribution and balancing in a Parallel Sysplex environment.

Glossary

7 **geocoder.** In DB2 Spatial Extender, a scalar function that translates existing data into data that can be understood in
7 spatial terms. For example, a geocoder that is supplied by Spatial Extender translates United States addresses into
7 instances of a spatial data type. Another geocoder might translate the identifier of a shelf in a warehouse into data
7 that identifies the location of that shelf in the warehouse.

7 **geodesic distance.** The shortest path between two points on the ellipsoidal shape of the Earth. This path might not
7 follow a line of constant latitude even though the two end points are at the same latitude.

7 **geographic coordinate system.** In DB2 Spatial Extender and DB2 Geodetic Extender, a reference system that uses
7 latitude and longitude to define locations on the surface of a sphere or spheroid.

7 **geographic feature.** An object on the surface of the Earth (such as a city or river), a space (such as a safety zone
7 around a hazardous site), or an event that occurs at a location (such as an auto accident that occurred at a particular
7 intersection).

| **geographic information system (GIS).** A complex of objects, data, and applications that is used to create and
| analyze spatial information about geographic features. See also “geographic feature.”

getpage. An operation in which DB2 Universal Database for z/OS and OS/390 accesses a data page.

| **ghost index.** An invisible index within the existing index object, created during the index create. It is not visible to
| users until it is fully created. See also “shadow index” on page 76.

| **GIS.** See “geographic information system .”

| **global catalog.** In a federated system, the database system catalog. The catalog contains information about objects in
| the federated database and information about objects at the data source. The catalog also contains information about
| the entire federated system. The information in the global catalog is used by the DB2 query optimizer to plan the
| best way to process SQL statements.

global lock. In DB2 Universal Database for z/OS and OS/390, a lock that provides concurrency control within and
among DB2 subsystems. The scope of the lock is across all DB2 subsystems of a data sharing group.

global lock contention. Conflicts on locking requests between different DB2 Universal Database for z/OS and
OS/390 members of a data sharing group when those members are trying to serialize shared resources.

| **global optimizer.** In a federated system, a feature of the DB2 SQL Compiler that analyzes the distributed queries
| and determines the most efficient way to run the query. The global optimizer evaluates queries based on resource
| cost. See “push-down processing” on page 65.

| **global record.** In SQL replication, the row in the register table that defines global replication characteristics for a
| particular instance of the Capture program.

| **global table lock.** A table lock that is acquired on all partitions of a table’s database partition group.

global transaction. A unit of work in a distributed transaction processing environment in which multiple resource
managers are required.

governor. See “resource limit facility” on page 71.

| **grant.** To give a privilege or authority to an authorization identifier.

graphic character. A DBCS character.

graphic string. A sequence of DBCS characters.

gross lock. In DB2 Universal Database for z/OS and OS/390, the shared, update, or exclusive mode locks on a
table, partition, or table space.

7 **group.** (1) A logical organization of users that have IDs according to activity or resource access authority. (2) In a
7 satellite environment, a collection of satellites that share characteristics such as the database configuration and the
7 application that runs on the satellite.

| **group buffer pool (GBP).** A coupling facility cache structure that is used by a data sharing group to cache data and
| to ensure that the data is consistent for all members. See also “cache structure” on page 9.

- 7 **group buffer pool duplexing.** In a z/OS or OS/390 environment, the ability to write data to two instances of a
7 group buffer pool structure: a primary group buffer pool and a secondary group buffer pool.
- 1 **grouping task.** A task in the Task Center that contains other tasks. You use a task to define task actions that depend
1 on the results of the tasks that the grouping task contains.
- 7 **group name.** In a z/OS or OS/390 environment, the XCF identifier for a data sharing group.
- 7 **group restart.** In a z/OS or OS/390 environment, a restart of at least one member of a data sharing group after the
7 loss of either locks or the shared communications area.
- 7 **group scope.** In a data sharing environment, the scope of a command that affects all members of a data sharing
7 group. See also “member scope” on page 52.
- GTF.** See “generalized trace facility” on page 37.
- GWAPI.** Domino Go Web server API.

H

- 1 **HACMP.** See “High Availability Cluster Multiprocessing (HACMP)” on page 40.
- 7 **HADR.** See “high availability disaster recovery” on page 40.
- handle.** (1) A variable that represents an internal structure within a software system. (2) A character string that is
created by an extender that is used to represent an image, audio, or video object in a table. A handle is stored for an
object in a user table and in administrative support tables. In this way, an extender can link the handle that is stored
in a user table with information about the object that is stored in the administrative support tables. (3) A binary value
7 that identifies a text document. A handle is created for each text document in a text column when that column is
7 *enabled* for use by the DB2 Net Search Extender.
- hash partitioning.** A partitioning strategy in which a hash function is applied to the partitioning key value to
determine the database partition to which the row is assigned.
- 1 **health.** The general condition or state of the database environment.
- 1 **Health Center.** The DB2 graphical interface that shows the overall state of the database environment and all current
1 alerts. From the Health Center, you can get details about alerts and recommended resolution actions.
- 1 **health indicator.** A measure of some aspect of the health of an object. Criteria are applied to the measurement to
1 determine healthiness, where the criteria applied depends on the type of health indicator as follows:
- 1 • **Threshold-based:** The measurement represents a statistic of the behavior of the object. Warning and alarm
1 threshold values set boundaries on the value of the statistic to define normal, warning, and alarm ranges.
 - 1 • **State-based:** The measurement represents two or more states, one of which is normal; and all others are considered
1 non-normal.
- 1 **health monitor.** An instance-level monitor that creates alerts based on a health indicator exceeding a threshold or
1 being in non-normal state. The monitor sends notifications to the notification log, and also sends emails and pages to
1 contacts on its notification list.
- 1 **health monitor alert.** An alert that is generated by the health monitor and is based on the type of health indicator,
1 which can be either threshold based (the health indicator value exceeds or falls below warning or alarm thresholds)
1 or state based (the health indicator value is a non-normal state).
- 7 **health notification contact list.** A set of contacts that receives notifications when health alerts occur.
- 1 **health snapshot.** Health data, retrieved from the database manager at a point in time, for a set of database objects.
- 7 **heap.** A logical grouping of memory that fulfills the needs of a particular component. For example, the utility heap
7 memory is used by DB2 utilities such as backup, restore, and load.

Glossary

2 **held query.** A query that was prevented from running by Query Patroller because its estimated cost is higher than a
2 Query Patroller threshold. The query is held until released from the held state by Query Patroller itself or by a user
2 with sufficient authority, such as an administrator. See also “intercepted query” on page 45 and “managed query” on
2 page 52.

1 **heterogeneous replication.** Replication between DB2 and non-DB2 relational databases. See also “federated system”
1 on page 35.

1 **hierarchical relationship category.** In the Information Catalog Center, a category of relationship types that are used
1 to connect objects that have a hierarchical relationship.

7 **High Availability Cluster Multiprocessing (HACMP).** An IBM application that links IBM pSeries servers or logical
7 partitions of pSeries servers into high-availability clusters to provide concurrent access to IT resources and the fault
7 resilience that is required for business-critical applications. See “failover” on page 34 and “fallback” on page 34.

7 **high availability disaster recovery (HADR).** A high-availability solution that is based on log shipping that provides
7 data availability on a standby system if a partial or complete site failure occurs on the primary system. Contrast with
7 “Q replication” on page 65.

7 **hiperspace.** In releases prior to DB2 UDB for z/OS Version 8, a storage space that contains up to 2 gigabytes of
7 contiguous virtual storage addresses that a program can use as a data buffer. Like a data space, a hiperspace can
1 hold user data; it does not contain common areas or system data. Unlike an address space or a data space, data in a
1 hiperspace is not directly addressable. To manipulate data in a hiperspace, you bring the data into the address space
1 in 4-KB blocks.

2 **historical analysis.** In Query Patroller, the activity of analyzing past usage of a data warehouse. This is done for
2 reasons including: management reporting, tuning, and optimizing Query Patroller thresholds.

7 **holdable result set.** A result set that is associated with a cursor that was created with the WITH HOLD clause. See
7 also “result set” on page 72.

1 **hole.** A row of the result table that cannot be accessed because a delete operation or an update operation has been
1 performed on that row. See also “delete hole” on page 26 and “update hole” on page 91.

7 **home address space.** In a z/OS or OS/390 environment, the area of storage that OS/390 currently recognizes as
7 *dispatched*.

hop. In APPN, a portion of a route that has no intermediate nodes. A hop consists of a single transmission group
connecting adjacent nodes.

host. In TCP/IP, any system that has at least one Internet address associated with it.

host computer. (1) In a computer network, a computer that provides services such as computation, database access,
and network control functions. (2) The primary or controlling computer in a multiple-computer installation.

host identifier. A name that is declared in the host program.

host language. Any programming language in which SQL statements can be embedded.

host node. In SNA, a subarea node that contains a system services control point (SSCP), for example, an IBM
System/390[®] computer with MVS and VTAM.

host program. A program written in a host language that contains embedded SQL statements.

host structure. In an application program, a structure that is referred to by embedded SQL statements.

host variable. In an application host program, a variable that is referred to by embedded SQL statements. Host
variables are programming variables in the application program and are the primary mechanism for transmitting
data between tables in the database and application program work areas.

7 **host variable array.** An array of elements, each of which corresponds to a value for a column. The dimension of the
7 array determines the maximum number of rows for which the array can be used.

1 **hot-spot update.** A series of repeated updates made to the same rows over a short period of time.

| **HTML.** See “Hypertext Markup Language.”

| **Hypertext Markup Language (HTML).** A markup language that conforms to the SGML standard and was designed primarily to support the online display of textual and graphical information that includes hypertext links. HTML is the primary markup language for documents on the Web.

I

7 **ICAPI.** Internet connection API.

7 **ICF.** In a z/OS or OS/390 environment, integrated catalog facility.

7 **IDCAMS.** In a z/OS or OS/390 environment, an IBM program that is used to process access method services commands. It can be invoked as a job or jobstep, from a TSO terminal or from within a user’s application program.

7 **IDCAMS LISTCAT.** In a z/OS or OS/390 environment, a facility for obtaining information from the access method services catalog.

identify. A request that an attachment service program (in an address space that is separate from DB2 Universal Database for z/OS and OS/390) issues through the MVS subsystem interface to inform DB2 Universal Database for z/OS and OS/390 of its existence and to initiate the process of becoming connected to DB2.

| **identity column.** A column that provides a way for DB2 to automatically generate a numeric value for each row that is inserted into the table. Identity columns are defined with the AS IDENTITY clause. A table can have no more than one identity column.

| **idle agent.** A database agent that currently does not have a database connection or an application attachment.

IFCID. In DB2 Universal Database for z/OS and OS/390, instrumentation facility component identifier.

IFI. In DB2 Universal Database for z/OS and OS/390, instrumentation facility interface.

IFI call. In DB2 Universal Database for z/OS and OS/390, an invocation of the instrumentation facility interface (IFI) by means of one of its defined functions.

7 **IFP.** In a z/OS or OS/390 environment, IMS Fast Path.

7 **IL.** See “intermediate language” on page 45.

ILU. See “independent logical unit” on page 42.

image copy. An exact reproduction of all or part of a table space. DB2 Universal Database for z/OS and OS/390 provides utility programs to make full image copies (to copy the entire table space) or incremental image copies (to copy only those pages that were modified since the last image copy).

7 **implicit connect.** A connection that is made to a database without a user ID or password.

7 **implicit privilege.** (1) A privilege that accompanies the ownership of an object, such as the privilege to drop that object. Different authority levels and database authorities can also provide implicit privileges on one or more objects. (2) A privilege on one or more data objects that are referenced by a package. Depending on how the package is bound to the database, the user can be granted implicit privileges on data objects that are referenced by the package when the package is executed. See also “privilege” on page 63 and “explicit privilege” on page 33.

| **import.** (1) To copy data from an external file, using formats such as PC/IXF, DEL, WSF or ASC, into database tables. See also “export” on page 33. (2) In the Information Catalog Center, to read the contents of a tag language file to initially populate the information catalog, change the information catalog contents, or copy the contents of another information catalog.

import metadata. The process of bringing metadata into the Data Warehouse Center, either dynamically (from the user interface) or in batch.

7 **IMS.** See “Information Management System” on page 43.

Glossary

- 1 **IMS attachment facility.** A DB2 Universal Database for z/OS and OS/390 subcomponent that lets users access DB2
1 Universal Database for z/OS and OS/390 from IMS. The IMS attachment facility receives and interprets requests for
1 access to DB2 databases by using exits that are provided by IMS subsystems.
- 1 **IMS DataPropagator.** A product that provides replication between IMS and DB2 databases. See also “replication” on
1 page 70.
- 1 **IMS DB.** Information Management System Database Manager.
- 1 **IMS TM.** Information Management System Transaction Manager.
- 7 **in-abort.** A status of a unit of recovery that occurs when DB2 Universal Database for z/OS and OS/390 fails after a
7 unit of recovery begins to be rolled back, but before the process is completed. DB2 Universal Database for z/OS and
7 OS/390 continues to back out the changes when the process restarts.
- 7 **in-commit.** A status of a unit of recovery that occurs when DB2 Universal Database for z/OS and OS/390 fails after
7 beginning its two-phase commit processing. When DB2 Universal Database for z/OS and OS/390 is restarted, this
7 status indicates that changes made to data are consistent.
- 1 **incremental backup.** A copy of all database data that has changed since the most recent successful full backup
operation. This is also known as a cumulative backup image, because a series of incremental backups taken over time
will each have the contents of the previous incremental backup image. The predecessor of an incremental backup
image is always the most recent successful full backup of the same object.
- 1 **incremental bind.** A process by which SQL statements are bound during the execution of an application process,
1 because they could not be bound during the bind process, and VALIDATE(RUN) was specified. See “bind” on page
1 8.
- independent.** In DB2 Universal Database for z/OS and OS/390, an object (row, table, or table space) that is neither
a parent nor a dependent of another object.
- 7 **independent auxiliary storage pool (IASP).** One or more storage units that are defined from the disk units or
7 disk-unit subsystems that make up addressable disk storage. An independent auxiliary pool contains objects, the
7 directories that contain the objects, and other object attributes such as authorization ownership attributes.
- 1 **independent logical unit (ILU).** A logical unit that is able to activate an LU-to-LU session without assistance from a
1 system services control point (SSCP). An ILU does not have an SSCP-to-LU session. See also “dependent logical unit”
1 on page 27 and “system services control point” on page 83.
- 1 **index.** A set of pointers that are logically ordered by the values of a key. Indexes provide quick access to data and
1 can enforce uniqueness on the rows in the table. When you request an index, the database manager builds the
1 structure and maintains it automatically. The index is used by the database manager to improve performance and
7 ensure uniqueness. See also “unique index” on page 90.
- 7 **index-controlled partitioning.** A type of partitioning in which partition boundaries for a partitioned table are
7 controlled by values that are specified on the CREATE INDEX statement. Partition limits are saved in the LIMITKEY
7 column of the SYSIBM.SYSINDEXPART catalog table.
- index file.** A file that contains indexing information used by the Video Extender to find a *shot* or an individual
frame in a video clip.
- index key.** The set of columns in a table that is used to determine the order of index entries.
- 1 **index partition.** The part of an index that is associated with a table partition at a given database partition. An index
1 defined on a table is implemented by multiple index partitions, one per table partition.
- index space.** In DB2 Universal Database for z/OS and OS/390, a page set that is used to store the entries of one
index.
- 7 **index specification.** In a federated system, a set of metadata about a data source object index that the query
7 optimizer uses to expedite the processing of distributed requests. When a nickname is created for a data source
7 object, the federated server gathers index information about that object and stores the information in the global
7 catalog.

indicator column. In DB2 Universal Database for z/OS and OS/390, a 4-byte value that is stored in a base table in place of a LOB column.

indicator variable. A variable that represents the null value in an application program. If the value for the selected column is null, a negative value is placed in the indicator variable.

| **individual privilege.** A privilege that is granted on a single data object. See also “privilege” on page 63.

7 **indoubt.** The status of a unit of recovery that occurs when the database manager fails after it finishes its phase 1
7 commit processing and before it starts phase 2. At emergency restart, the status of the unit of recovery is indoubt
7 until the commit coordinator indicates to the database manager whether the unit of recovery is to be committed or
7 rolled back.

indoubt resolution. The process of resolving the status of an indoubt logical unit of work to either the committed or the rollback state.

| **indoubt transaction.** A transaction in which one phase of a two-phase commit completes successfully but the
| system fails before a subsequent phase can complete.

| **infix operator.** An operator that is used in comparison expressions. See also “comparison operator” on page 14.

7 **inflight.** A status of a unit of recovery that occurs when DB2 Universal Database for z/OS and OS/390 fails before
7 its unit of recovery completes phase 1 of the commit process. When DB2 Universal Database for z/OS and OS/390 is
7 restarted, it backs out the updates of any units of recovery that have inflight status.

| **informational configuration parameter.** A type of configuration parameter that holds information that cannot be
| modified. See also “configurable configuration parameters” on page 15 and “configurable online configuration
| parameters” on page 15.

7 **informational constraint.** A rule used by the SQL compiler to improve query performance without requiring
7 additional data verification. See also “check constraint” on page 11, “referential constraint” on page 68, and “unique
7 constraint” on page 90.

7 **informational message.** In Q replication and event publishing, a message about the status of the Q Capture
7 program, a Q subscription, or an XML publication.

| **information catalog.** A collection of metadata, managed by the Information Catalog Center, that contains descriptive
| data (business metadata) that helps users identify and locate data and information that is available to them in the
| organization. An information catalog also contains some technical metadata.

| **Information Catalog Center.** A DB2 graphical interface for organizing, maintaining, finding, and using business
| information. The Information Catalog Center is a part of the Information Catalog Manager.

| **Information Catalog Manager.** A set of tools designed to help organize, maintain, find, and use business
| information. The Information Catalog Manager is comprised of the Information Catalog Center, the Manage
| Information Catalog wizard, and the Information Catalog Manager sample catalogs. A Web version of the
| Information Catalog Center is also available.

| **Information Catalog Manager application program interface (API).** A set of Java classes that can be used to write
| programs that read, create, and update the metadata that is stored in the information catalog.

7 **Information Management System (IMS).** Any of several system environments available with Database Manager
7 and Transaction Manager, capable of managing complex databases and terminal networks.

| **information source.** An item of data or information, such as a table or chart, that is represented by an Information
| Catalog Center object.

inheritance. The passing of class resources or attributes from a parent class downstream in the class hierarchy to a child class.

initialization fullselect. The first fullselect in a recursive common table expression that gets the direct children of the initial value from the source table.

7 **inline SQL PL.** A subset of SQL procedural language that can be used in SQL functions, triggers, and dynamic
7 compound statements.

Glossary

inner join. A join method in which a column that is not common to all of the tables being joined is dropped from the resultant table. See “join” on page 46. See also “outer join” on page 58.

7 **inoperative package.** A package that cannot be used because one or more user-defined functions or procedures on
7 which the package depends were dropped. Such a package must be explicitly rebound. See also “invalid package” on
page 46.

1 **inoperative trigger.** A trigger that depends on an object that has been dropped or made inoperative or on a
privilege that has been revoked. See also “trigger” on page 88.

7 **inoperative view.** A view that is not usable because a privilege on an underlying table is revoked, a table, alias, or
7 function is dropped, the superview becomes inoperative, or another view on which the view is dependent is either
7 dropped or becomes inoperative.

1 **input relationship type.** In the Information Catalog Center, a relationship type that is used to connect objects that
1 transform to their input data resource. See “transformation relationship category” on page 87. See also “relationship
1 type” on page 69.

1 **insensitive cursor.** A cursor that is not sensitive to inserts, updates, or deletes that are made to the underlying rows
1 of a result table after the result table has been materialized. See also “sensitive cursor” on page 75.

insert rule. A condition enforced by the database manager that must be met before a row can be inserted into a
table.

7 **insert trigger.** A trigger that is activated upon the event of an insert operation on the base table. See “trigger” on
7 page 88 and “trigger activation” on page 88.

1 **installation program.** A program that prepares a software package to run on the computer. During installation, a
1 component of the setup program is commonly copied to the disk drive to allow the user to customize the program’s
1 default settings.

1 **installation verification scenario.** A sequence of operations that runs the main DB2 Universal Database functions
7 and tests whether DB2 Universal Database was correctly installed. The operations include installing the DB2
7 Universal Database server, installing the DB2 Universal Database client, configuring a client-to-server connection, and
7 issuing a connect statement from the client to the server.

instance. (1) See also “database manager instance” on page 21. (2) A logical DB2 extender server environment. You
can have several instances of DB2 extender servers on the same system, but only one instance for each DB2 instance.

1 **instance-owning partition.** The first database-partition server that is installed in a partitioned database environment.

1 **instantiable structured type.** A structured type that can be used for creating database objects. A structured type that
1 is not instantiable cannot be used for creating database objects; however, such a type can be used to define subtypes
1 which, in turn, can be instantiable.

7 **instead of trigger.** A trigger that is associated with a single view and is activated by an insert, update, or delete
7 operation on the view, and that defines how to propagate the insert, update, or delete operation on the view to the
7 underlying tables of the view. See also “trigger” on page 88, “before trigger” on page 7, and “after trigger” on page
7 2.

instrumentation facility component identifier (IFCID). In DB2 Universal Database for z/OS and OS/390, a value
that names and identifies a trace record of an event that can be traced. As a parameter on the START TRACE and
MODIFY TRACE commands, it specifies that the corresponding event is to be traced.

instrumentation facility interface (IFI). A programming interface that enables programs to obtain online trace data
about DB2 Universal Database for z/OS and OS/390, to submit DB2 Universal Database for z/OS and OS/390
commands, and to pass data to DB2 Universal Database for z/OS and OS/390.

1 **interactive SQL.** A set of SQL statements that is provided through an interface such as the Command Center or
1 command line processor. These statements are processed as dynamic SQL statements. For example, an interactive
1 SELECT statement can be processed dynamically using the DECLARE CURSOR, PREPARE, DESCRIBE, OPEN,
1 FETCH, and CLOSE statements.

- 1 **Interactive System Productivity Facility (ISPF).** In a z/OS or OS/390 environment, an IBM licensed program that
1 provides interactive dialog services. Users can perform most DB2 Universal Database tasks interactively through ISPF
1 panels.
- 2 **intercepted query.** A query that Query Patroller intercepts from another application and that can be either managed
2 or not managed by Query Patroller. Managed queries are a subset of intercepted queries. See also “managed query”
2 on page 52 and “held query” on page 40.
- inter-DB2 R/W interest.** In DB2 Universal Database for z/OS and OS/390, a property of data in a table space,
index, or partition that has been opened by more than one member of a data sharing group and that has been
opened for writing by at least one of those members.
- 1 **intermediate database server.** The target of a request from a local application or a remote application requester that
is being forwarded to another database server because the object doesn’t exist on the target database server. The
remote request is forwarded transparently to another database server if the object referred to by the three-part name
does not refer to the local location. See also “database server” on page 21.
- 7 **intermediate language (IL).** A type of compiled bytecode that is interpreted by the .NET Framework common
7 language runtime. Source code from all .NET compatible languages compiles to IL bytecode. See also “common
7 language runtime” on page 14 and “assembly” on page 5.
- intermediate network node.** In APPN, a node that is part of a route between an origin logical unit (OLU) and a
destination logical unit (DLU) but that neither contains the OLU or the DLU nor serves as the network server for
either the OLU or DLU.
- 7 **internal CCD table.** A CCD table that is not a registered replication source and therefore cannot be subscribed to.
7 An internal CCD table is identified by the CCD_OWNER and CCD_TABLE columns for the row of the associated
7 registered replication source. Contrast with “external CCD table” on page 33. See also “consistent-change-data table”
7 on page 16.
- 1 **internal resource lock manager (IRLM).** A DB2 Universal Database for z/OS and OS/390 component that allows
serial access to data. DB2 requests locks from IRLM to ensure data integrity when applications, utilities, and
commands attempt to access the same data.
- 1 **internationalization.** In software engineering, the process of producing a product that is independent of any
1 particular language, script, culture, and coded character set. See also “Unicode” on page 89.
- Internet Protocol (IP).** A protocol that is used to route data from its source to its destination in an Internet
environment. See also “Transmission Control Protocol/Internet Protocol” on page 88.
- 7 **Internet suite of protocols.** A set of protocols that were developed for use on the Internet and published through
7 the Internet Engineering Task Force (IETF).
- 7 **inter-partition parallelism.** A single database operation (for example index creation) that is run in parallel across the
7 partitions of a partitioned database. See also “intra-partition parallelism.”
- 1 **interprocess communication (IPC).** A mechanism of an operating system that allows processes to communicate with
1 each other within the same computer or over a network.
- 1 **inter-query parallelism.** The ability of a database to accept queries from multiple applications. Each query runs
independently of the others, but DB2 runs all of them at the same time. See also “intra-query parallelism.”
- 7 **interval timing.** In SQL replication, the process of controlling how frequently a replication subscription cycle runs.
7 Contrast with “event timing” on page 32.
- 1 **intra-partition parallelism.** The subdivision of a single database operation (for example, index creation) into
1 multiple parts, which are then executed in parallel within a single database partition. See also “inter-partition
1 parallelism.”
- 1 **intra-query parallelism.** The ability to process parts of a single query at the same time using either “intra-partition
parallelism,” “inter-partition parallelism,” or both.

Glossary

invalid package. A package that depends on an object that has been dropped. See also “inoperative package” on page 44.

invariant character set. (1) A character set, such as the syntactic character set, whose code point assignments do not change from code page to code page. (2) A minimum set of characters that is available as part of all character sets. See also “syntactic character set” on page 83.

I/O parallelism. See “parallelism” on page 59.

IP. See “Internet Protocol” on page 45.

IP address. The unique address that specifies the location of each device or workstation in an intranet or the Internet. For example, 9.67.97.103 is an IP address.

IRLM. See “internal resource lock manager” on page 45.

ISAPI. Microsoft® Internet Server API.

isolation level. (1) A security feature that determines how data is locked from other processes while it is being accessed. See also “repeatable read” on page 70, “read stability” on page 67, “cursor stability” on page 19, and “uncommitted read (UR)” on page 89. (2) An attribute that defines the degree to which an application process is isolated from other concurrently executing application processes.

ISPF. See “Interactive System Productivity Facility” on page 45.

ISPF/PDF. In a z/OS or OS/390 environment, Interactive System Productivity Facility/Program Development Facility.

J

Java archive. A compressed file format in which all of the resources that are required to install and run a Java program are stored in a single file. Commonly known as a JAR file.

Java Database Connectivity (JDBC). An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC provides a call-level API for SQL-based database access.

JCL. See “job control language.”

JDBC driver. A program included with database management systems to support the JDBC standard access between the databases and Java applications.

JES. See “Job Entry Subsystem.”

JFS. See “journaled file system” on page 47.

job control language (JCL). A command language that is used to identify a job to an operating system and to describe the job’s requirements.

Job Entry Subsystem (JES). An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs.

job scheduler. A program that is used to automate certain tasks for running and managing database jobs.

join. An SQL relational operation that allows retrieval of data from two or more tables based on matching column values. See also “broadcast join” on page 8, “collocated join” on page 13, “full outer join” on page 36, “inner join” on page 44, “left outer join” on page 48, “outer join” on page 58, and “right outer join” on page 72.

joined table. An intermediate result table that is the result of either an inner join or an outer join.

journal. (1) For iSeries systems, a system object that identifies the objects being journaled, the current journal receiver, and all the journal receivers on the system for the journal. The system-recognized identifier for the object type is *JRN. See also “journal receiver” on page 47. (2) The destination pages from which you can view all available historical information about task history, database history, PM alerts, messages, and the notification log.

7 **journal code.** On iSeries systems, a 1-character code in a journal entry that identifies the category of the journal
 7 entry. For example, F identifies an operation on a file, R identifies an operation on a record, and so on. See also
 7 “journal entry type.”

7 **journal entry type.** On iSeries systems, a two-character field in a journal entry that identifies the type of operation
 7 of a system-generated journal entry or the type of journal entry of a user-generated journal entry. For example, PT is
 7 the entry type for a write operation. See also “journal code.”

| **journaled file system (JFS).** The native file system in the AIX operating system.

| **journal receiver.** For iSeries systems, a system object that contains journal entries added when events occur that are
 | journaled, such as changes to a database file, changes to other journaled objects, or security-relevant events. The
 | object type is *JRNRCV. See also “journal” on page 46.

K

| **Kerberos.** A network authentication protocol that is designed to provide strong authentication for client/server
 | applications by using secret-key cryptography. See also “Kerberos ticket.”

| **Kerberos ticket.** A transparent application mechanism that transmits the identity of an initiating principal to its
 | target. A simple ticket contains the principal’s identity, a session key, a timestamp, and other information, which is
 | sealed using the target’s secret key.

| **key.** A column or an ordered collection of columns that is identified in the description of a table, index, or
 | referential constraint. The same column can be part of more than one key.

7 **key-sequenced data set (KSDS).** In a z/OS or OS/390 environment, a VSAM file or data set whose records are
 7 loaded in key sequence and controlled by an index.

7 **key-value based partitioning strategy.** A strategy for assigning rows in a table to database partitions. Rows are
 7 assigned based on the values of the partitioning key columns. See also “partitioning key” on page 60.

7 **keyword.** (1) One of the predefined words of a programming language, artificial language, application, or
 7 command. (2) A name that identifies an option that is used in an SQL statement.

KSDS. See “key-sequenced data set.”

L

labeled duration. A number that represents a duration of years, months, days, hours, minutes, seconds, or
 microseconds.

7 **large object (LOB).** A data type that contains a sequence of bytes that can range in size from 0 bytes to 2 gigabytes
 7 less 1 byte. There are three types of large objects: binary large objects (binary), character large objects (single-byte
 7 character or mixed), and double-byte character large objects (double-byte character). See also “character large object”
 7 on page 11, “binary large object” on page 7, and “double-byte character large object” on page 30.

| **large table space.** A table space that can store only long string or large object (LOB) or index data.

latch. A DB2 Universal Database for z/OS and OS/390 internal mechanism for controlling concurrent events or the
 use of system resources.

| **latency.** The time that is needed for updates that are made to a source to replicate to a target.

7 **LCID.** In a z/OS or OS/390 environment, log control interval definition.

LDS. See “linear data set” on page 48.

| **leaf page.** A page that contains pairs of keys and record identifiers and that points to actual data. See also “nonleaf
 | page” on page 56.

Glossary

| **left outer join.** The result of a join operation that includes the matched rows of both tables that are being joined and
| that preserves the unmatched rows of the first table. See “join” on page 46. See also “right outer join” on page 72 and
| “full outer join” on page 36.

LEN node. See “low-entry networking node” on page 51.

length attribute. A value associated with a string that represents the declared fixed length or maximum length of
the string.

linear data set (LDS). In an OS/390 environment, a VSAM data set that contains data but no control information. A
linear data set can be accessed as a byte-addressable string in virtual storage.

| **link.** The action that DB2 Data Links Manager takes to control a file that is referenced in a table that contains a
7 DATALINK column. A file can be linked as the result of such database actions as an SQL UPDATE, INSERT,
7 IMPORT, or LOAD statement.

linkage editor. A computer program for creating load modules from one or more object modules or load modules
by resolving cross-references among the modules and, if necessary, adjusting addresses.

| **linked file.** In DB2 Data Links Manager, a file that is referenced in a table’s DATALINK column that is defined with
| LINK CONTROL. To guarantee referential integrity, a linked file is maintained under the control of the DLFF
| component.

| **link-edit.** In DB2 Universal Database for z/OS and OS/390, the action of creating a loadable computer program by
| using a linkage editor.

| **linked relationship type.** In the Information Catalog Center, a relationship type that is used to connect two or more
| objects in an information catalog. Objects in a linked relationship are peers, rather than a parent-child relationship.
| For example, in the sample information catalog that is included with the Information Catalog Center, the object called
| **CelDial Sales Information** is linked with objects that describe CelDial advertisements for the year. See also
| “relationship type” on page 69.

| **list.** A type of object, which DB2 utilities can process, that identifies multiple table spaces, multiple index spaces, or
| both. A list is defined with the LISTDEF utility control statement.

| **list prefetch.** An access method that takes advantage of prefetching even in queries that do not access data
sequentially. A list prefetch is done by scanning the index and collecting record identifiers before any data pages are
accessed. These record identifiers are then sorted, and data is prefetched using this list.

7 **list structure.** In a z/OS or OS/390 environment, a coupling facility structure that provides the ability to share and
7 manipulate data as elements of a queue.

7 **little endian.** A format for storage or transmission of binary data in which the least significant bit (or byte) is placed
7 first.

| **load authority.** An access level that gives LOAD utility or AutoLoader utility privileges to load data into tables.

load copy. A backup image of data that was previously loaded and that can be restored during rollforward recovery.

load module. A program unit that is suitable for loading into main storage for execution. A load module is the
output of a linkage editor.

7 **load phase.** In Q replication, the stage where a target table is loaded with data from a source table so that the two
7 tables are synchronized. The load phase can be automatic or manual.

LOB. See “large object” on page 47.

LOB locator. A mechanism that allows an application program to manipulate a large object (LOB) value in the
database system. A LOB locator is a simple token value that represents a single LOB value. An application program
retrieves a LOB locator into a host variable and can then apply SQL functions to the associated LOB value using the
locator.

LOB lock. In DB2 Universal Database for z/OS and OS/390, a lock on a LOB value.

LOB table space. In DB2 Universal Database for z/OS and OS/390, a table space that contains all the data for a particular LOB column in the related base table.

- 7 **local.** Pertaining to a device, file, or system that is accessed from a user's system without the use of a
7 communication line. See also "remote" on page 70.
- 7 **local catchup state.** The state that a high availability disaster recovery (HADR) standby database enters when it has
7 been started and then attempts to read logs that reside locally in its log path or by way of a user exit program. After
7 they are read, the logs are replayed on the standby database.
- 7 **local database.** A database that is located on the system in use. See also "remote database" on page 70.

| **local database directory.** A directory where a database physically resides. Databases that are displayed in the local
database directory are located on the same node as the system database directory.

7 **locale.** (1) In DB2 Universal Database for z/OS and OS/390, the definition of a subset of a user's environment that
7 combines characters that are defined for a specific language and country or region, and a CCSID. (2) A collection of
7 processing variables that are used to specify how a process runs. Computer locales include the conventions for a
7 specific language and culture, with appropriate date and time formatting, character classification, sorting, and text
7 handling.

| **local lock.** A lock that provides intra-DB2 concurrency control, but not inter-DB2 concurrency control; its scope is a
| single DB2 Universal Database for z/OS and OS/390 subsystem.

| **local subsystem.** The unique relational database management system to which the user or application program is
| directly connected (in the case of DB2 Universal Database for z/OS and OS/390, by one of the DB2 Universal
| Database for z/OS and OS/390 attachment facilities).

local table lock. A table lock that is acquired only on a single database partition.

local update. An update to the base table, not to the replica.

7 **location alias.** Another name by which a database server identifies itself in the network. Applications can use this
7 name to access a DB2 database server.

location name. (1) The name by which DB2 Universal Database for z/OS and OS/390 refers to a particular DB2
1 subsystem in a network of subsystems. (2) The unique name of a database server. An application uses the location
1 name to access a DB2 database server.

location path. A subset of the abbreviated syntax of the location path defined by XPath. A sequence of XML tags to
7 identify an XML element or attribute. It is used in extracting user-defined functions to identify the subject to be
7 extracted, and it is used by user-defined functions in DB2 Net Search Extender to identify the search criteria.

locator. See "LOB locator" on page 48.

| **locator variable.** A host variable that contains the locator that represents a LOB value on the application server.

lock. (1) A means of serializing events or access to data. (2) A means of preventing uncommitted changes made by
one application process from being perceived by another application process and for preventing one application
7 process from updating data that is being accessed by another process. A lock ensures the integrity of data by
7 preventing concurrent users from accessing inconsistent data.

| **lock duration.** The interval over which a DB2 Universal Database for z/OS and OS/390 lock is held. For example,
| locks on LOBs are taken when they are needed and are usually released at commit.

lock escalation. The response that occurs when the number of locks issued for one agent exceeds the limit specified
in the database configuration; the limit is defined by the *maxlocks* configuration parameter. During a lock escalation,
7 locks are freed by converting locks on rows of a table into one lock on a table. This process is repeated until the limit
7 is no longer exceeded.

lock mode. A representation for the type of access that concurrently running programs can have to a resource that a
DB2 Universal Database for z/OS and OS/390 lock is holding.

lock object. The resource that is controlled by a DB2 Universal Database for z/OS and OS/390 lock.

Glossary

lock promotion. The process of changing the size or mode of a DB2 Universal Database for z/OS and OS/390 lock to a higher level.

lock size. The amount of data that is controlled by a DB2 Universal Database for z/OS and OS/390 lock on table data; the value can be a row, a page, a LOB, a partition, a table, or a table space.

lock structure. In DB2 Universal Database for z/OS and OS/390, a coupling facility data structure that is composed of a series of lock entries to support shared and exclusive locking for logical resources.

7 **log.** (1) A record of events. (2) A collection of records that describes the events that occur in a system and indicates
7 the sequence of the events. See also “database log” on page 20.

7 **log file.** (1) The file where the log of events is recorded. (2) A file that is produced by the Information Catalog
Center when it imports a tag language file into the information catalog. This file records the times and dates when
| the import process started and stopped and any error information for the process.

log head. The oldest log record in the active log.

| **logical agent.** An agent that represents the client or application connection.

logical claim. In DB2 Universal Database for z/OS and OS/390, a claim on a logical partition of a nonpartitioning index.

| **logical data group.** A collection of data elements that gather database system monitoring information at a specific
| scope of database activity. The snapshot monitor and event monitor each have their own sets of logical data groups.
| See “data element” on page 22.

| **logical data modeling.** The process of documenting the comprehensive business information requirements in an
| accurate and consistent format. Data modeling is the first step in designing a database.

logical drain. In DB2 Universal Database for z/OS and OS/390, a drain on a logical partition of a nonpartitioning index.

logical index partition. In DB2 Universal Database for z/OS and OS/390, the set of all keys that reference the same data partition.

logical lock (L-lock). In DB2 Universal Database for z/OS and OS/390, the lock type that transactions use to control intra-DB2 and inter-DB2 data concurrency between transactions. See also “physical lock” on page 61.

logical operator. A keyword that specifies how multiple search conditions are to be evaluated (AND, OR) or if the logical sense of a search condition is to be inverted (NOT).

| **logical page list (LPL).** In DB2 Universal Database for z/OS and OS/390, a list of pages that are in error and that
| cannot be referenced by applications until the pages are recovered. The page is in logical error, even though the
| actual media (coupling facility or DASD) might not contain any errors. Usually, a connection to the media has been
| lost.

logical partition. (1) In DB2 Universal Database for z/OS and OS/390, a set of key or RID pairs in a nonpartitioning index that are associated with a particular partition. (2) In a partitioned database environment, a database partition server on a processor that has more than one database partition server assigned to it

logical recovery pending (LRECP). In DB2 Universal Database for z/OS and OS/390, the state in which the data and the index keys that refer to the data are inconsistent.

7 **logical server.** (1) In replication, on Linux, UNIX, and Windows, a DB2 database. (2) On z/OS, a subsystem that is
7 running DB2 Universal Database.

7 **logical table.** In Q replication, all of the copies of one table that are distributed across servers in bidirectional or
7 peer-to-peer replication. If a user replicates two logical tables, then copies of both tables are distributed across the
7 servers.

7 **logical unit (LU).** An access point through which a user or application program accesses the SNA network to
7 communicate with another user or application program. See also “LU name” on page 51.

logical unit 6.2 (LU 6.2). The LU type that supports sessions between two applications using APPC.

logical unit of work (LUW). The processing that a program performs between synchronization points.

7 **logical unit of work identifier (LUWID).** In a z/OS or OS/390 environment, a name that uniquely identifies a
7 thread within a network. This name consists of a fully qualified logical unit network name, a logical unit of work instance number, and a logical unit of work sequence number.

log initialization. The first phase of restart processing during which DB2 Universal Database for z/OS and OS/390 attempts to locate the current end of the log.

log record. A record of an update to a database performed during a unit of work. This record is written after the log tail of the active log.

| **log record header (LRH).** In DB2 Universal Database for z/OS and OS/390, a prefix for a logical record that
| contains control information. Only the first segment contains the entire LRH; later segments include only the first two
| fields. When a specific log is needed for recovery, all segments are returned and presented together as if the record
| were stored continuously.

log record sequence number (LRSN). A number that DB2 Universal Database for z/OS and OS/390 generates and associates with each log record. The LRSN is also used for page versioning. The LRSNs that a particular DB2 Universal Database for z/OS and OS/390 data sharing group generates form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the data sharing group.

7 **log table.** A table created by the DB2 Net Search Extender that contains information about which text documents are
7 to be indexed.

log tail. The log record that was written most recently in an active log.

| **log truncation.** In DB2 Universal Database for z/OS and OS/390, a process by which an explicit starting relative
| byte address is established. This RBA is the point at which the next byte of log data is to be written.

| **long string.** A variable-length string whose maximum length is greater than 254 bytes.

| **long table space.** See “large table space” on page 47.

low-entry networking node (LEN node). A type 2.1 node that supports independent LU protocols but does not support CP to CP sessions. It can be a peripheral node attached to a boundary node in a subarea network, an end node attached to an APPN network node in an APPN network, or a peer-connected node directly attached to another LEN node or APPN end node.

LPL. See “local page list” on page 50.

LRECP. See “logical recovery pending” on page 50.

LRH. See “log record header.”

LRSN. See “log record sequence number.”

LU. See “logical unit” on page 50.

7 **LU name.** In a z/OS or OS/390 environment, the name by which VTAM refers to a node in a network. See also
7 “location name” on page 49.

LU 6.2. See “logical unit 6.2” on page 50.

| **LU type.** The classification of a logical unit in terms of the specific subset of SNA protocols and options that it
| supports for a given session. Specifically, the values allowed in the session activation request and the usage of data
| stream controls, function management headers, request unit parameters, sense data values, and presentation services
| protocols such as those associated with function management headers.

LUW. See “logical unit of work.”

LUWID. See “logical unit of work identifier.”

Glossary

M

- 7 **maintenance window.** A user-defined time period for running only required automatic maintenance activities. See
7 also “automatic maintenance” on page 6.
- 2 **managed query.** A query that is subject to the Query Patroller thresholds and parameters that control how a query
2 is handled, for example, whether it is allowed to run, queued, or rejected. See also “intercepted query” on page 45
2 and “held query” on page 40.
- 7 **manual load.** In Q replication, a load process in which the user loads data into a target table and then notifies the
7 replication program when the table is loaded. See also “automatic load” on page 6.
- mapped conversation.** In APPC, a conversation between two transaction programs (TPs) using the APPC mapped
conversation API. In typical situations, end-user TPs use mapped conversation, and service TPs use basic
conversations. Either type of program can use either type of conversation. See also “basic conversation” on page 7.
- masking character.** A character used to represent optional characters at the front, middle, and end of a search term.
Masking characters are normally used for finding variations of a term in a precise index.
- 1 **mass delete.** The deletion of all rows of a table.
- 1 **massively parallel processing (MPP).** The coordinated execution of a single request either by multiple
single-processor computers in a shared-nothing environment (in which each computer has its own memory and
disks); or by symmetric multiprocessor (SMP) computers (in which multiple processors in each computer share
memory and disks). Both environments require that all computers are linked together in a high-speed network.
- 1 **master table.** In SQL replication, specifically in update-anywhere replication, the original source table for data in the
1 replica table. If replication conflict detection is enabled, changes made to the master table are retained, whereas
1 changes made to the replica table are rejected. See also “update-anywhere replication” on page 90, “replica table” on
1 page 70, and “conflict detection” on page 15.
- materialize.** (1) In DB2 Universal Database for z/OS and OS/390, to put rows from a view or nested table
expression into a work file for additional processing by a query. (2) To place a LOB value into contiguous storage.
Because LOB values can be very large, DB2 Universal Database for z/OS and OS/390 avoids materializing LOB data
until doing so becomes absolutely necessary.
- 1 **materialized query table.** A table whose definition is based on the result of a query and whose data is in the form
1 of precomputed results that are taken from the table, or tables, that the materialized query table definition is based
1 on.
- 1 **MBCS.** See “multibyte character set” on page 54.
- 2 **MDC table.** See “multidimensional clustering table” on page 54.
- member name.** The XCF identifier for a particular DB2 Universal Database for z/OS and OS/390 subsystem in a
data sharing group.
- 7 **member scope.** In a data sharing environment, the scope of a command that affects only the DB2 databases for
7 which it is issued. See also “group scope” on page 39.
- member state.** In DB2 Universal Database for z/OS and OS/390, the state of the DB2 member (subsystem) of the
data sharing group.
- 7 **menu.** A displayed list of available functions for selection by the user.
- 1 **merge.** To update and insert new content into a table.
- 1 **message processing program (MPP).** An IMS online program that can access DB2 Universal Database for z/OS and
1 OS/390 databases, full-function databases, data entry databases, and main storage databases.
- 1 **metadata.** Data that describes the characteristics of stored data; descriptive data. For example, the metadata for a
1 database table might include the name of the table, the name of the database that contains the table, the names of the
1 columns in the table, and the column descriptions, either in technical terms or business terms. Database catalogs and
1 information catalogs contain metadata.

1 **metadata publication process.** A process created by the Data Warehouse Center that contains all the steps to keep
1 published metadata synchronized with the original metadata.

7 **method.** A database object that is created by running the CREATE METHOD statement and encapsulated logic that
7 provides behavior for structured types. A method can be implemented as an SQL method or an external method. See
7 also “scalar method” on page 74, “SQL method” on page 79, and “external method” on page 34.

migration. (1) The process of moving data from one computer system to another without converting the data. (2)
Installation of a new version or release of a program to replace an earlier version or release.

1 **mixed-character string.** A string that contains a mixture of single-byte and multibyte characters. Synonym for
1 *mixed-data string*.

7 **mobile client.** The node, usually a mobile computer, where the mobile enabler is located.

mode. In the Data Warehouse Center, the stage of development of a step, such as development, test, or production.

mode name. (1) In APPC, the name used by the initiator of a session to designate the characteristics desired for the
session, such as message length limits, sync point, class of service within the transport network, and session routing
7 and delay characteristics. (2) In a z/OS or OS/390 environment, a VTAM name for the collection of physical and
7 logical characteristics and attributes of a session.

7 **MODEENT.** In a z/OS or OS/390 environment, a VTAM macro instruction that associates a logon mode name with
7 a set of parameters that represent session protocols. A set of MODEENT macro instructions defines a logon mode
7 table.

modeled statistics. Statistics for a database object that might or might not be referenced in an SQL statement, yet
currently exist in an explain model. The object does not need to currently exist in the database.

1 **modeling database.** In a z/OS or OS/390 environment, a DB2 database that is created on a workstation that models
1 a DB2 Universal Database subsystem in the z/OS and OS/390 environment. A modeling database can be used for
1 indexing and query optimization.

modify lock. In DB2 Universal Database for z/OS and OS/390, an L-lock or P-lock with a MODIFY attribute. A list
of these active locks is kept at all times in the coupling facility lock structure. If the requesting subsystem fails, that
subsystem’s modify locks are converted to retained locks.

7 **Monitor control server.** In replication, a database that contains the Monitor control tables, which store information
7 about alert conditions that the Replication Alert Monitor will monitor.

7 **monitor element.** A data structure that is used by the system monitor to store information about a particular aspect
7 of the database system status. Data elements collect data for one or more logical data groups. Each monitor element
7 collects one of the following specific types of data: counter, gauge, watermark, textual information, or timestamp. See
7 also “logical data group” on page 50.

7 **monitoring task.** In the Activity Monitor, a set of reports and filter settings that collect specific snapshot data to
7 troubleshoot applications or statements, or to tune queries for optimal use of database resources.

7 **Monitor qualifier.** In replication, a case-sensitive character string that identifies an instance of the Replication Alert
7 Monitor.

7 **monitor switch.** A database manager parameter that is manipulated by the user to control the type of information
7 and the quantity of information that is returned in performance snapshots.

1 **monotonic decreasing expression.** An expression or function used to derive a generated column that has the
1 property that for every possible pair of values x_1 and x_2 , if $x_2 > x_1$ then $fn(x_2) < fn(x_1)$.

1 **monotonic increasing expression.** An expression or function used to derive a generated column that has the
1 property that for every possible pair of values x_1 and x_2 , if $x_2 > x_1$ then $fn(x_2) > fn(x_1)$.

1 **monotonic nondecreasing expression.** An expression or function used to derive a generated column that has the
1 property that for every possible pair of values x_1 and x_2 , if $x_2 > x_1$ then $fn(x_2) \geq fn(x_1)$.

1 **monotonic nonincreasing expression.** An expression or function used to derive a generated column that has the
1 property that for every possible pair of values x_1 and x_2 , if $x_2 > x_1$ then $fn(x_2) \leq fn(x_1)$.

Glossary

MPP. See “message processing program” on page 52 or “massively parallel processing” on page 52.

| **MQT.** See “materialized query table” on page 52.

MTO. In OS/390 environment, master terminal operator.

multibyte character set (MBCS). A set of characters in which each character is represented by 1 or more bytes. Contrast with “double-byte character set” on page 30 and “single-byte character set” on page 77. See also “ASCII” on page 5, “single-byte character set” on page 77, “EBCDIC” on page 31, and “Unicode” on page 89.

7 **multidimensional.** In the DB2 OLAP Server, pertaining to a method of referencing data through three or more
7 dimensions. An individual data value in a fact table is the intersection of one member from each dimension. See also
| “business dimension” on page 9 and “dimension” on page 28.

| **multidimensional analysis.** The process of assessing and evaluating an enterprise on more than one level.

multidimensional database. In the DB2 OLAP Server, a nonrelational database into which you copy relational data
| for OLAP analysis. See also “relational cube” on page 69.

2 **multidimensional clustering (MDC) table.** A table whose data is physically organized into blocks along one or
2 more dimensions, or clustering keys, specified in the ORGANIZE BY DIMENSIONS clause.

7 **multidirectional replication.** In Q replication, a replication configuration that includes peer-to-peer or bidirectional
7 replication.

| **multiple logical partition configuration.** In a partitioned database environment, a configuration in which more than
| one database partition server is assigned to a computer, and these database partition servers are recorded in the same
| db2nodes.cfg file.

| **multisite update.** Distributed relational database processing in which data is updated in more than one location
| within a single unit of work.

multitasking. A mode of operation that provides for concurrent performance or interleaved execution of two or more tasks.

| **multi-tier replication.** In SQL replication, a replication configuration in which changes are replicated from a
| replication source in one database to a replication target in another database, and changes from this replication target
| are replicated again to a replication target in another database.

must-complete. A state during DB2 Universal Database for z/OS and OS/390 processing in which the entire operation must be completed to maintain data integrity.

| **Multiple Virtual Storage (MVS).** The primary operating system used on IBM mainframe computers. This operating
| system manages large amounts of memory and disk space.

N

7 **namespace.** In XML, a uniform resource identifier (URI) that provides a unique name to associate with all the
7 elements and type definitions in a schema.

NAU. See “network addressable unit” on page 55.

7 **near synchronous mode.** In high availability disaster recovery (HADR), the synchronization mode in which the
7 primary database considers a transaction committed when it receives a message from the standby database
7 confirming that the log data was received and written to the main memory of the standby system. See “peer state”
7 on page 61.

negotiable lock. In DB2 Universal Database for z/OS and OS/390, a lock whose mode can be downgraded, by agreement among contending users, to be compatible to all. A physical lock is an example of a negotiable lock.

7 **nested savepoint.** A savepoint that is included or positioned within another savepoint. Nested savepoints allow an
7 application to have multiple levels of savepoints active at a time, and allows the application to rollback to any active
7 savepoint as desired.

| **nested table expression.** A fullselect in a FROM clause (surrounded by parentheses).

network address. An identifier for a node in a network.

network addressable unit (NAU). The origin or the destination of information transmitted by the path control network. An NAU can be a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). See also “network name.”

7 **network identifier (NID).** In a z/OS or OS/390 environment, the network identifier that is assigned by IMS or
7 CICS, or if the connection type is RRSAF, the z/OS and OS/390 RRS unit of recovery identifier (URID).

| **Network Information Service (NIS/NIS+).** On AIX, a central record of passwords, nodes, and related data that can
| be used with the DB2 Administration Server in the administration of user and group names.

| **network name.** In SNA, a symbolic name by which users refer to a network addressable unit (NAU), a link station,
| or a link.

| **NETWORK netid.** The identifier of the SNA network where the remote LU resides. This network ID is a string of
| one to eight characters that follows the naming convention for SNA.

| **network node (NN).** In APPN, a node on the network that provides distributed directory services, topology
| database exchanges with other APPN network nodes, and session and routing services. See also “Advanced
| Peer-to-Peer Networking” on page 2.

network node server. An APPN network node that provides network services for its local logical units and adjacent end nodes.

network-qualified name. The name by which an LU is known throughout an interconnected SNA network. A network-qualified name consists of a network name identifying the individual subnetwork, and a network LU name. Network-qualified names are unique throughout an interconnected network. Also known as the *network-qualified LU name*, or *fully qualified LU name*.

network services. The services within network addressable units that control network operation through SSCP-to-SSCP, SSCP-to-PU, SSCP-to-LU, and CP-to-CP sessions.

| **nickname.** (1) In a federated system, an identifier that is used in a query to refer to an object at a data source. The
| objects that nicknames identify are referred to as *data source objects*. Examples of data source objects include tables,
7 views, synonyms, table-structured files, and search algorithms. (2) A name that is defined in DB2 Information
7 Integrator to represent a physical database object (such as a table or stored procedure) in a non-DB2 relational
7 database.

NID. See “network identifier.”

| **NIS/NIS+.** See “Network Information Service.”

NN. See “network node.”

| **node.** (1) In communications, an end point of a communications link, or a junction common to two or more links in
| a network. Nodes can be processors, communication controllers, cluster controllers, terminals, or workstations. Nodes
| can vary in routing and other functional capabilities. (2) In hardware, a uniprocessor or symmetric multiprocessor
| (SMP) computer that is part of a clustered system or a massively parallel processing (MPP) system. For example,
| RS/6000® SP™ is an MPP system that consists of nodes connected by a high-speed network. (3) An obsolete term for
| database partition. See “database partition” on page 21.

| **node directory.** A directory that contains information that is necessary to establish communications from a client
| workstation to all applicable database servers.

| **nodegroup.** An obsolete term for database partition group. See “database partition group” on page 21.

7 **noncomplete CCD table.** In SQL replication, a CCD table that is initially empty and has rows appended to it as
7 changes are made to the replication source. Contrast with “complete CCD table” on page 15. See also
7 “consistent-change-data table” on page 16.

noncondensed attribute. A table attribute that indicates that the table contains a history of changes to the data, not current data. A table that has this attribute set includes more than one row for each key value.

Glossary

- | **noncondensed CCD table.** In SQL replication, a CCD table that can contain more than one row for each key value. These duplicate rows represent the history of changes for the values in rows of a table. Contrast with “condensed CCD table” on page 15. See also “consistent-change-data table” on page 16.
- | **noncumulative backup image.** See “delta backup” on page 27.
- | **non-DB2 relational database server.** An Informix database server or a relational database server from a vendor other than IBM.
- | **nondelimited ASCII format.** A file format that is used to import data. Nondelimited ASCII is a sequential ASCII file with row delimiters used for data exchange with any ASCII product.
- | **nonleaf page.** A page that contains keys and page numbers of other pages in the index (either leaf or nonleaf pages). Nonleaf pages never point to actual data. See also “leaf page” on page 47.
- 7 **nonpartitioned index.** An index that is not physically partitioned. Both partitioning indexes and secondary indexes can be nonpartitioned.
- 7 **nonpartitioned secondary index (NPSI).** A secondary index that is not partitioned. See also “secondary index” on page 74.
- 7 **nonpartitioning index.** See “secondary index” on page 74.
- | **nonscrollable cursor.** A cursor that can be moved only in a forward direction. Nonscrollable cursors are sometimes called forward-only cursors or serial cursors. See also “scrollable cursor” on page 74.
- | **normalization.** The process of restructuring a data model by reducing its relations to their simplest forms. It is a key step in the task of building a logical relational database design. Normalization helps avoid redundancies and inconsistencies in your data. An entity is normalized if it meets a set of constraints for a particular normal form (first normal form, second normal form, and so on). See also “denormalization” on page 27 and “repeating group” on page 70.
- | **not deterministic function.** In DB2 Universal Database for z/OS and OS/390, a user-defined function whose result is not solely dependent on the values of the input arguments. Successive invocations with the same argument values can produce a different answer. This type of function is sometimes called a “variant function” on page 92. Contrast with “deterministic function” on page 27.
- 7 **not fenced.** Pertaining to a type, or characteristic, of a procedure, user-defined function, or federated wrapper that is defined to run in the database manager process. When this type of object is run (using the not fenced clause), the database manager is not protected from changes made by this object. See also “fenced” on page 35.
- | **notification log.** See “administration notification log” on page 2.
- | **notification process.** A process created by the Data Warehouse Center that contains all the steps created for notification when a step completes.
- | **not variant function.** Synonym for “deterministic function” on page 27. See also “variant function” on page 92.
- 7 **NPSI.** See “nonpartitioned secondary index.”
- 7 **NRE.** In a z/OS or OS/390 environment, network recovery element.
- | **NTFS.** One of the native file systems in the Windows NT and later operating environments such as Windows 2000.
- | **NULL.** (1) In the C programming language, a single character that denotes the end of the string. (2) In SQL, the value denoting the absence of data for a particular row and column.
- | **null.** A value that indicates the absence of information.
- 7 **nullable.** The condition in which a value for a column, function parameter, or result can have an absence of a value.
- | **NULLIF.** In DB2 Universal Database for z/OS and OS/390, a scalar function that evaluates two passed expressions, and returns either NULL if the arguments are equal or the value of the first argument if they are not.

1 **null indicator.** A column (by byte position) in a nondelimited ASCII file that contains the null indicator flag for the
 7 data being loaded into a table column. The value of the null indicator flag can be any valid positive integer. See also
 7 “null indicator flag.”

1 **null indicator flag.** A one-byte character that is contained in a “null indicator” column of a nondelimited ASCII file.
 When the load process looks at each data row, the null indicator flag indicates whether the data in the column that is
 defined by the start and end positions is null.

7 **null terminated host variable.** In DB2 Universal Database for z/OS and OS/390, a varying-length host variable in
 7 which the end of the data is indicated by the presence of a null terminator.

7 **null terminator.** In C, the value that indicates the end of a string. For EBCDIC, ASCII, and Unicode UTF-8 strings,
 7 the null terminator is a single-byte value (X'00'). For Unicode UTF-16 and UCS-2 (wide) strings, the null terminator is
 7 a double-byte value (X'0000').

null value. A parameter position for which no value is specified.

O

OASN. See “origin application schedule number” on page 58.

1 **object.** (1) Anything that can be created or manipulated with SQL—for example, tables, views, indexes, or packages.
 1 (2) In object-oriented design or programming, an abstraction that consists of data and operations associated with that
 1 data. (3) In the Information Catalog Center, an item that represents a unit or distinct grouping of information. Each
 1 Information Catalog Center object identifies and describes information but does not contain the actual information.
 1 For example, an object can provide the name of a report, list its creation date, and describe its purpose.

1 **object type.** (1) A categorization or grouping of object instances that share similar behaviors and characteristics. (2)
 1 In the Information Catalog Center, a classification for objects. An object type is used to reflect a type of business
 1 information, such as a table, report, or image.

OBID. In DB2 Universal Database for z/OS and OS/390, data object identifier.

1 **occasionally connected.** In SQL replication, a replication configuration that contains target servers that are not
 1 always connected to the network. This configuration allows users to connect to a primary data source for a short
 1 time to synchronize their local database with the data at the source.

ODBC. See “Open Database Connectivity” on page 58.

ODBC driver. A driver that implements ODBC function calls and interacts with a data source.

7 **offline backup.** A backup of the database or table space that is made while the database or table space is not being
 7 accessed by applications. During an offline backup, the backup database utility acquires exclusive use of the database
 1 until the backup is complete. See also “online backup.”

7 **offline maintenance.** Maintenance activities that can occur only when user access to a database is interrupted. See
 7 also “online maintenance” on page 58.

7 **offline restoration.** A restoration of a copy of a database or table space from a backup. The restore database utility
 has exclusive use of the database until the restore is completed. See also “online restoration” on page 58.

OLAP. See “online analytical processing.”

old structure. See “primary group buffer pool” on page 63.

1 **online analytical processing (OLAP).** In the DB2 OLAP Server, a multidimensional, multi-user, client server
 1 computing environment for users who need to analyze consolidated enterprise data in real time. OLAP systems
 1 feature zooming, data pivoting, complex calculations, trend analysis, and modeling.

online backup. A backup of the database or table space that is made while the database or table space is being
 accessed by other applications. See also “offline backup.”

7 **online index creation.** The process of creating a new index while allowing the underlying table and any previously
 7 existing indexes to be read and updated by concurrent transactions.

Glossary

1 **online index reorganization.** Reorganizing indexes on a table while allowing the table and existing indexes to be
1 read and updated by concurrent transactions.

online maintenance. Maintenance activities that can occur while users are connected to a database. See also “offline maintenance” on page 57.

7 **online restoration.** A restoration of a copy of a table space while applications are able to access the tables in other
7 table spaces. See also “offline restoration” on page 57.

7 **Open Database Connectivity (ODBC).** An application program interface (API) that allows access to database
7 management systems by using callable SQL, which does not require the use of an SQL preprocessor. The ODBC
7 architecture allows users to add modules, called *database drivers*, that link the application to their choice of database
7 management systems at run time.

1 **operator.** (1) An action that must be performed on data, or the output from a table or an index, when the access
2 plan for an SQL statement is executed. (2) In Query Patroller, a person who has a subset of administrator authorities
2 as defined in their operator profile.

1 **operational data.** Data that is used to run the day-to-day operations of an organization.

operand. An entity on which an operation is performed.

optimized SQL text. SQL text, produced by the Explain facility, that is based on the query actually used by the optimizer to choose the access plan. This query is supplemented and rewritten by the various components of the SQL compiler during statement compilation. The text is reconstructed from its internal representation, and differs from the original SQL text. The optimized statement produces the same result as the original statement.

1 **option.** In the Information Catalog Center tag language, a parameter of the ACTION tag that defines the action to
1 be performed on objects or object types in the information catalog when the tag language file is imported.

7 **ordinary identifier.** An uppercase letter followed by zero or more characters, each of which is an uppercase letter, a
7 digit, or the underscore character. An ordinary identifier should not be a reserved word.

ordinary token. A numeric constant, an ordinary identifier, a host identifier, or a keyword.

7 **origin application schedule number (OASN).** In a z/OS or OS/390 environment with IMS, a 4-byte number that is
1 assigned sequentially to each IMS schedule since the last cold start of IMS. The OASN is used as an identifier for a
1 unit of work. In an 8-byte format, the first 4 bytes contain the schedule number and the last 4 bytes contain the
7 number of IMS sync points (*commit points*) during the current schedule. The OASN is part of the network identifier
7 (NID) for an IMS connection.

originating task. In DB2 Universal Database for z/OS and OS/390, the primary agent in a parallel group that receives data from other execution units (referred to as *parallel tasks*) that are executing portions of the query in parallel.

7 **orphaned contact.** A contact that exists in the health notification contact list but is not defined in the contact list that
7 is stored on the system that is specified by the CONTACT_HOST configuration parameter of the DB2 administration
7 server. See also “contact” on page 16.

1 **outer join.** (1) A join method in which a column that is not common to all of the tables being joined becomes part of
1 the resultant table. (2) The result of a join operation that includes the matched rows of both tables that are being
joined and preserves some or all of the unmatched rows of the tables that are being joined. See “join” on page 46. See
also “inner join” on page 44, “full outer join” on page 36, “left outer join” on page 48, and “right outer join” on page
72.

1 **outline.** In the DB2 OLAP Server, the structure that defines all elements of a database within the DB2 OLAP Server.
For example, an outline contains definitions of dimensions, members, and formulas.

output file. A database or device file that is opened with the option to allow the writing of records.

1 **output relationship type.** In the Information Catalog Center, a relationship type that is used to connect objects that
1 transform to their output data resource. See “transformation relationship category” on page 87. See also “relationship
1 type” on page 69.

overflow record. (1) An updated record that is too large to fit on the page it is currently stored in. The record is copied to a different page and its original location is replaced with a pointer to the new location. (2) On an indirectly addressed file, a record whose key is randomized to the address of a full track or to the address of a home record. (3) In the event monitor, a record that is inserted in the event monitor data stream to indicate that records were discarded because the named pipe was full and records were not processed in time. An overflow record indicates how many records were discarded.

overloaded function name. A function name for which multiple functions exist within a function path or schema. Those within the same schema must have different signatures.

ownership privilege. Control privilege that allows all privileges for the owned data object. See also “privilege” on page 63.

P

package. (1) A control structure produced during program preparation that is used to execute SQL statements. (2) In Java programming, a program statement that defines the location of a Java class within the directory structure, or library, of a Java application.

package list. In DB2 Universal Database for z/OS and OS/390, an ordered list of package names that can be used to extend an application plan.

package name. The name of an object that is created by BIND, PRECOMPILE, or REBIND command. The object is a bound version of a database request module (DBRM). The name consists of a location name, a collection ID, a package ID, and a version ID.

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

page. (1) A block of storage within a table or index whose size is 4096 bytes (4 KB). (2) A unit of storage within a table space (4 KB, 8 KB, 16 KB, or 32 KB) or index space (4 KB). In a table space, a page contains one or more rows of a table. In a LOB table space, a LOB value can span more than one page, but no more than one LOB value is stored on a page. (3) In a notebook in the graphical interface, a predefined display image that typically provides fields and controls that help users accomplish tasks.

page set. In a z/OS or OS/390 environment, a table space or an index space that consists of a collection of VSAM data sets.

page set recovery pending (PSRCP). In DB2 Universal Database for z/OS and OS/390, a restrictive state of an index space in which the entire page set must be recovered.

panel. A formatted display of information on a screen.

parallel group. In a z/OS or OS/390 environment, a set of consecutive operations that run in parallel and have the same number of parallel tasks.

parallel I/O processing. (1) A form of I/O processing in which DB2 Universal Database for z/OS and OS/390 initiates multiple concurrent requests for a single user query and performs I/O processing concurrently (in parallel) on multiple data partitions. (2) The process of reading from or writing to two or more I/O devices at the same time to reduce response time.

parallelism. The ability to perform multiple database operations at the same time. See also “inter-partition parallelism” on page 45, “intra-partition parallelism” on page 45, and “I/O parallelism” on page 46.

parallel session. In SNA, two or more concurrently active sessions between the same two logical units. Each session can have different session parameters. See “session” on page 76.

Parallel Sysplex. A set of z/OS or OS/390 systems that communicate and cooperate with each other through multisystem hardware components and software services to process customer workloads.

parallel task. In a z/OS or OS/390 environment, the execution unit that is dynamically created to process a query in parallel.

Glossary

parameterized data type. A data type that can be defined with a specific length, scale, or precision. String and decimal data types are parameterized.

parameter marker. A question mark (?) that appears in a statement string of a dynamic SQL statement. The question mark can appear where a host variable might appear if the statement string was a static SQL statement.

| **parameter-name.** A long identifier that names a parameter that can be referenced in a procedure or user-defined function.

parent key. A primary key or unique key that is used in a referential constraint. The values of a parent key determine the valid values of the foreign key in the constraint.

7 **parent lock.** For explicit hierarchical locking in DB2 Universal Database for z/OS and OS/390, a lock that is held on
7 a resource that might have child locks that are lower in the hierarchy. A parent lock is usually the table space or the
7 partition intent locks. See also “child lock” on page 12.

parent row. A row that has at least one dependent row.

parent table. A table that is a parent in at least one referential constraint.

| **parent table space.** In DB2 Universal Database for z/OS and OS/390, a table space that contains a parent table. See
| also “dependent table space” on page 27.

| **partial declustering.** In a partitioned database environment, the storage of table data on a named subset of database
| partitions (database partition group), rather than on all database partitions for the database.

7 **participant.** In a z/OS or OS/390 environment, an entity other than the commit initiator that takes part in the
7 commit process. Synonym for *agent* in SNA.

7 **partition.** In a z/OS or OS/390 environment, a portion of a page set. Each partition corresponds to a single,
independently extendable data set. Partitions can be extended to a maximum size of 1, 2, or 4 gigabytes, depending
on the number of partitions in the partitioned page set. All partitions of a given page set have the same maximum
size.

| **partition-compatible join.** A join where all of the rows that are joined reside in the same database partition. See also
| “join” on page 46 and “collocated join” on page 13.

| **partitioned database.** A database with two or more database partitions. Each database partition stores a subset of
| table data for each table that resides on it. See “database partition” on page 21.

7 **partitioned data set (PDS).** In a z/OS or OS/390 environment, a data set in direct-access storage that is divided into
7 partitions, which are called members. Each partition can contain a program, part of a program, or data. Synonym for
program library.

7 **partitioned index.** An index that is physically partitioned. Both partitioning indexes and secondary indexes can be
7 partitioned.

7 **partitioned page set.** In a z/OS or OS/390 environment, a partitioned table space or an index space. Header pages,
7 space map pages, data pages, and index pages refer to data only within the scope of the partition.

7 **partitioned table space.** In a z/OS or OS/390 environment, a table space that is subdivided into parts (based on
7 index key range), each of which can be processed independently by utilities.

| **partitioning agent.** Within AutoLoader, the process used to create the partition files for loading. This is done by
| splitting or partitioning the input file.

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

partitioning key. (1) An ordered set of one or more columns in a given table. For each row in the table, the values in the partitioning key columns are used to determine on which database partition the row belongs. (2) In replication, an ordered set of one or more columns in a given table. For each row in the source table, the values in the partitioning key columns are used to determine in which target table the row belongs.

partitioning map. A vector of partition numbers that maps a partitioning map index to database partitions in the database partition group.

partitioning map index. A number that is assigned to a hash partition or range partition.

7 **partition pruning.** The removal from consideration of inapplicable partitions through setting up predicates in a
7 query on a partitioned table to access only certain partitions to satisfy the query.

partner logical unit (LU). (1) In SNA, the remote participant in a session. (2) An access point in the SNA network that is connected to the local DB2 Universal Database for z/OS and OS/390 subsystem by way of a VTAM conversation.

| **pass-through.** In a federated system, pertaining to a special DB2 Universal Database session used to submit SQL
| statements directly to DBMSs using the SQL dialect associated with that data source. Use a pass-through session
| when you want to perform an operation that is not possible with DB2 SQL/API, or to perform actions not supported
| by SQL.

| **path.** (1) In an operating system, the route through a file system to a specific file. (2) In a network environment, the
| route between any two nodes. See also “SQL path” on page 79.

PCT. In CICS, program control table.

PDS. See “partitioned data set” on page 60.

7 **peer state.** In high availability disaster recovery (HADR), the state during which the primary database ships new
7 log pages from its in-memory log buffer to the standby database when the same log pages are written to the log disk
7 of the primary database.

7 **peer-to-peer communication.** The communication between two SNA logical units (LUs) that is not managed by a
7 DB2 family database; commonly used when referring to LU 6.2 nodes.

| **peer to peer relationship category.** In the Information Catalog Center, a category of relationship types that are used
| to connect objects that have a peer relationship.

| **peer-to-peer replication.** A replication configuration between logical tables in which updates at any table are
| replicated to the other tables, and convergence is maintained. Peer-to-peer replication can have two servers or three
| or more servers. In SQL replication, contrast with “update-anywhere replication” on page 90. See also “multi-tier
| replication” on page 54.

performance metrics. A collection of all performance variables that belong to the same database object.

performance snapshot. Performance data for a set of database objects that is retrieved from the database manager at a point in time.

7 **performance variable.** A statistic that is derived from performance data that is obtained from the database manager.
7 The expression for this variable can be defined by the user.

phantom row. A table row that can be read by application processes that are executing with any isolation level except repeatable read. When an application process issues the same query multiple times within a single unit of work, additional rows can appear between queries because of the data being inserted and committed by application processes that are running concurrently.

physical claim. In DB2 Universal Database for z/OS and OS/390, a claim on an entire nonpartitioning index.

physical consistency. In DB2 Universal Database for z/OS and OS/390, the state of a page that is not in a partially changed state.

physical drain. In DB2 Universal Database for z/OS and OS/390, a drain on an entire nonpartitioning index.

physical lock (P-lock). A lock type that DB2 Universal Database for z/OS and OS/390 acquires to provide consistency of data that is cached in different DB2 Universal Database for z/OS and OS/390 subsystems. Physical locks are used only in data sharing environments. See also “logical lock” on page 50.

physical lock contention. In DB2 Universal Database for z/OS and OS/390, conflicting states of the requesters for a physical lock. See also “negotiable lock” on page 54.

Glossary

physically complete. In DB2 Universal Database for z/OS and OS/390, the state in which the concurrent copy process is completed and the output data set has been created.

7 **physical unit (PU).** The component that manages and monitors the resources (such as attached links and adjacent
7 link stations) that are associated with a node, as requested by an system services control point (SSCP) through an
7 SSCP-to-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the physical
unit, resources of the node such as attached links. This term applies to types 2.0, 4, and 5 nodes only. See also
“control point” on page 17.

7 **piece.** In a z/OS or OS/390 environment, a data set of a nonpartitioned page set.

plan. See “application plan” on page 4.

plan allocation. The process of allocating DB2 Universal Database for z/OS and OS/390 resources to a plan in preparation to execute it.

plan name. In DB2 Universal Database for z/OS and OS/390, the name of an application plan.

plan segmentation. In DB2 Universal Database for z/OS and OS/390, the dividing of each plan into sections. When a section is needed, it is independently brought into the EDM pool.

P-lock. See “physical lock” on page 61.

7 **PLT.** See “program list table” on page 64.

7 **plug-in.** A dynamically loadable library that DB2 Universal Database uses to carry out user-written actions that
7 involve the database.

7 **point-in-time table.** In SQL replication, a type of replication target table whose content matches all or part of a
7 source table, with an added column that identifies the approximate time when the particular row was inserted or
7 updated at the source system.

point of consistency. A point in time when all the recoverable data that a program accesses is consistent. The point of consistency occurs when updates, insertions, and deletions are either committed to the physical database or rolled back. See also “rollback” on page 72 and “commit point” on page 14.

policy. See “CFRM policy” on page 11.

7 **pool.** See “heap” on page 39.

| **populate.** To add object types, objects, or metadata to the Information Catalog Center.

| **possibly uncommitted.** A state assigned by the index manager to an index key where the completion of the COMMIT of the insertion or deletion of that key cannot be determined.

postponed abort UR. In DB2 Universal Database for z/OS and OS/390, a unit of recovery that was in-flight or in-abort, was interrupted by system failure or cancellation, and did not complete backout during restart.

| **power user.** A person who has special privileges to perform some object management tasks, such as creating and
| updating objects. See “user” on page 91.

PPT. (1) In CICS, processing program table. (2) In OS/390, program properties table.

| **precision.** In numeric data types, the total number of binary or decimal digits, excluding the sign. The sign is
| considered positive if the value of a number is zero.

precompile. To process programs that contain SQL statements before they are compiled. SQL statements are replaced with statements that will be recognized by the host language compiler. The output from a precompile process includes source code that can be submitted to the compiler and used in the bind process.

predicate. An element of a search condition that expresses or implies a comparison operation.

prefetch. To read data before, and in anticipation of, its use.

prefetch processing. In DB2 Universal Database for z/OS and OS/390, an operation in which data is read by one of the following mechanisms: sequential prefetch or list sequential prefetch (also referred to as list prefetch).

| **prefix.** In a DB2 Data Links Manager environment, an absolute path in a DLFS under which linked files are stored.

| **prepare.** (1) To convert an SQL statement from text form to an executable form, by submitting it to the SQL compiler. (2) The first phase of a two-phase commit process in which all participants are requested to prepare for commit.

| **prepared SQL statement.** In SQL, a named object that is the executable form of an SQL statement that is processed by the PREPARE statement.

| **primary authorization ID.** The authorization identifier that is used to identify the application process to DB2 Universal Database for z/OS and OS/390.

7 **primary database.** In high availability disaster recovery (HADR), the main database, which is accessed by
7 applications. Applications apply updates to the primary database, and those updates are propagated on the standby
7 database by using log shipping.

| **primary group buffer pool.** For a duplexed group buffer pool, the DB2 Universal Database for z/OS and OS/390
| structure that is used to maintain the coherency of cached data. This structure is used for page registration and
| cross-invalidation. The z/OS or OS/390 equivalent is *old* structure. See also “secondary group buffer pool” on page
| 74.

| **primary index.** In DB2 Universal Database for z/OS and OS/390, an index that enforces the uniqueness of a primary key.

| **primary key.** (1) A unique key that is part of the definition of a table. A primary key is the default parent key of a
| referential constraint definition. It is a column or combination of columns that uniquely identifies a row in a table. (2)
7 In a federated system, a unique key that is part of the definition of a nickname and that the optimizer uses to
7 improve query performance. This key is not validated when operations such as insert and update are performed.

| **primary log.** A set of one or more log files used to record changes to a database. Storage for these files is allocated in advance. See also “secondary log” on page 74.

7 **primary reintegration.** The process in which the original primary database can rejoin a high availability disaster
7 recovery (HADR) pair after a failover. The original primary database can rejoin only as the new secondary database
7 in the HADR pair.

| **principal.** An entity that can communicate securely with another entity. In Kerberos, principals are represented as entries in the Kerberos registry database and include users, servers, computers, and others.

| **private connection.** A communications connection that is specific to DB2 Universal Database for z/OS and OS/390.
| For example, when the application server is a DB2 Universal Database subsystem, DB2 Universal Database private
| connections are allocated as necessary to support references to objects at other DB2 Universal Database subsystems.
| Like SQL connections, DB2 Universal Database private connections are initially in the held state and can be placed in
| the release pending status.

| **private protocol access.** A method of accessing distributed data by which you can direct a query to another DB2 Universal Database system. See also “DRDA access” on page 30.

| **private protocol connection.** A DB2 Universal Database private connection of the application process. For example,
| if the first phase of your application program uses DB2 Universal Database private protocol access and the second
| phase uses DRDA access, then open DB2 Universal Database private protocol connections from the first phase might
| cause a CONNECT operation to fail in the second phase. See also “private connection.”

7 **privilege.** (1) The right to access a specific database object in a specific way. Privileges are controlled by users with
7 SYSADM (system administrator) authority or DBADM (database administrator) authority or by creators of objects.
7 For example, privileges include rights to create, delete, and retrieve data from tables. (2) In DB2 Universal Database
for z/OS and OS/390, the capability of performing a specific function, sometimes on a specific object. See also
“explicit privilege” on page 33, “implicit privilege” on page 41, and “authority” on page 5.

Glossary

- 7 **privilege set.** For the predefined SYSADM ID, the set of all possible privileges. For any other authorization
| identifier, the set of all privileges that are recorded for that identifier in the DB2 Universal Database for z/OS and
| OS/390 catalog.
- procedure.** See “stored procedure” on page 80.
- 7 **procedure call.** See “call” on page 9.
- process.** (1) In the Data Warehouse Center, a series of steps, which commonly operates on source data, that changes
data from its original form into a form conducive to decision support. A Data Warehouse Center process commonly
7 consists of one or more sources, one or more steps, and one or more targets. (2) See “application process” on page 4.
- 7 **processing program table (PPT).** A table that defines the application programs and BMS maps that can run under
7 CICS.
- 7 **profile.** A set of characteristics that defines how Query Patroller handles a submitter’s queries or what tasks an
7 operator can perform. See also “operator” on page 58 and “submitter” on page 81.
- 7 **program list table (PLT).** A CICS control table that contains a list of programs that can run as a group during CICS
7 startup or shutdown, and can be enabled and disabled as a group by a single CEMT transaction.
- | **programs object type.** An object type that identifies and describes applications that are capable of processing the
| information that is described by Information Catalog Center objects.
- 7 **projected coordinate system.** In DB2 Spatial Extender, a reference system that defines locations on a planar surface.
- | **promote.** In SQL replication, to copy replication definitions for subscription sets or registered sources from one
| database to another database, without registering the sources again or creating the subscription sets again.
- | **propagation.** A process in which groups of configuration parameters are updated and take effect at different rates.
- 7 **property.** A characteristic or attribute that describes a unit of information.
- | **property name.** The 254-byte descriptive name of a property that is displayed in the Information Catalog Center
| user interface.
- 7 **protected conversation.** In a z/OS or OS/390 environment, a VTAM conversation that supports two-phase commit
7 flows.
- | **pruning.** In replication, the task of removing obsolete data from replication control tables or log files that are used
| by the Capture, Q Capture, Apply, and Q Apply programs.
- PSRCP.** See “page set recovery pending” on page 59.
- | **pseudo deleted.** A key that is marked as deleted but has not yet been physically removed from the index page.
- | **pseudo delete index.** An index in which deletion of keys is not done until sometime after the deletion is committed.
| See also “type 2 indexes” on page 89.
- | **pseudo empty.** An index page is pseudo empty if all the keys on the page are marked as pseudo deleted. See also
| “pseudo deleted.”
- public authority.** The authority for an object granted to all users.
- 7 **publishing queue map.** In event publishing, an object that includes a send queue for sending messages and settings
7 for how a Q Capture program processes all transactions that use the send queue. See also “replication queue map”
7 on page 71 and “queue map” on page 66.
- PU.** See “physical unit” on page 62.
- | **pull configuration.** In SQL replication, a configuration in which the Apply program runs at the target server. The
| Apply program pulls updates from the source server to apply them to the target. Contrast with “push configuration”
| on page 65.

push configuration. In SQL replication, a configuration in which the Apply program runs at the source server or a replication server other than the target server. The Apply program pushes updates from the source server to apply them to the target. Contrast with “pull configuration” on page 64.

push-down processing. In a federated system, the processing of segments of a query at a data source instead of at the federated server.

PU type. In SNA, the classification of a physical unit according to the type of node on which it resides.

Q

Q Apply latency. In Q replication, the amount of time for a transaction to be applied to a target table after the Q Apply program gets the transaction from a receive queue.

Q Apply program. In Q replication, a program that reads transactions from a receive queue and applies those changes to one or more target tables or passes the changes to a stored procedure.

Q Apply schema. In Q replication, the identifier for a Q Apply program and its control tables.

Q Apply server. In Q replication, a database or subsystem on which the control tables for the Q Apply program are located and where the Q Apply program runs. It contains one or more sets of the control tables that store information about target tables and other replication definitions.

QBIC. See “Query by Image Content” on page 66.

Q Capture latency. In Q replication, an approximate measure of how current a Q Capture program is in reading the DB2 Universal Database recovery log. Q Capture latency measures the time between when a Q Capture program saves performance data and the timestamp of the last committed transaction that the program read in the log when it saved the data. For example, if the Q Capture program saved performance data at 10 a.m. and the timestamp of the last committed transaction was 9:59 a.m., the Q Capture latency is one minute.

Q Capture program. In Q replication and event publishing, a program that reads the DB2 Universal Database recovery log to capture changes made to DB2 Universal Database source tables and transmits the changes by using one or more send queues.

Q Capture schema. In Q replication, the identifier for a Q Capture program and its control tables.

Q Capture server. In Q replication and event publishing, a database or subsystem on which the control tables for the Q Capture program are located and where the Q Capture program runs. The Q Capture server contains one or more sets of the control tables that store information about Q subscriptions and XML publications and other replication or publishing definitions.

Q Capture transaction latency. In Q replication, the time from when a Q Capture program reads the commit statement for a transaction in the DB2 Universal Database recovery log, to the time when the Q Capture program puts the message that contains the transaction on a send queue.

Q replication. A replication solution that uses WebSphere MQ message queues for high-volume, low-latency replication.

Q subscription. In Q replication, an object that identifies a mapping between a source table and target table or stored procedure and specifies what changes are replicated. Contrast with “XML publication” on page 94.

Q subscription group. In Q replication, the group of Q subscriptions that are involved in replicating the same logical tables.

quantified predicate. A predicate that compares a value with a set of values.

quantile. A subgroup that is created when a group is divided into equal, ordered parts.

query. (1) A request for information from the database based on specific conditions, such as a request for a list of all customers in a customer table whose balance is greater than \$1000. (2) In DB2 Universal Database for z/OS and OS/390, a component of certain SQL statements that specifies a result table.

Glossary

query block. In DB2 Universal Database for z/OS and OS/390, the part of a query that is represented by one of the FROM clauses. Each FROM clause can have multiple query blocks, depending on how DB2 Universal Database for z/OS and OS/390 internally processes the query.

1 **Query by Image Content (QBIC).** A capability that is provided by the Image Extender that allows users to search images by their visual characteristics, such as average color and texture.

2 **query class.** In Query Patroller, a mechanism that allows queries to be grouped and run according to their size.
2 Query classes allow the flow of queries on a database to be controlled, ensuring that resources are shared among
2 queries in the different size groupings (query classes).

7 **query controller.** The server component of the Query Patroller system, which works with the DB2 Universal
7 Database server to manage queries.

1 **query CP parallelism.** In DB2 Universal Database for z/OS and OS/390, parallel execution of a single query, which
1 is accomplished by using multiple tasks. See also “Sysplex query parallelism” on page 83.

query I/O parallelism. In DB2 Universal Database for z/OS and OS/390, parallel access to data, which is accomplished by triggering multiple I/O requests within a single query.

1 **query optimization class.** A set of query rewrite rules and optimization techniques for compiling queries.

1 **query optimizer.** A component of the SQL compiler that chooses an access plan for a data manipulation language
1 statement by modeling the execution cost of many alternative access plans and choosing the one with the minimal
1 estimated cost.

2 **Query Patroller Center.** The graphical interface for administering Query Patroller. You can use the Query Patroller
2 Center to monitor and control the flow of queries, manage users, and view query information.

7 **query status.** In Query Patroller, the state of processing that a managed query is in: initial (intercepted), running,
7 done, held, queued, canceled, aborted, rejected, or unknown. See also “managed query” on page 52.

2 **query submission preferences.** In Query Patroller, a set of characteristics for a submitter that the submitter can
2 view and update. See also “submitter” on page 81.

7 **queue.** A WebSphere MQ object that holds messages for message queuing applications. A queue is owned and
7 maintained by a queue manager.

7 **queue latency.** In Q replication and event publishing, the time from when the Q Capture program puts a transaction
7 on a send queue until the Q Apply program gets the transaction from the receive queue.

7 **queue map.** In Q replication and event publishing, an object that links queues and defines how the Q Capture and
7 Q Apply programs process messages that use the queues. See also “publishing queue map” on page 64 and
7 “replication queue map” on page 71.

2 **queue priority.** In Query Patroller, a numerical value that represents the priority assigned to a query when a query
2 is submitted.

1 **queued sequential access method (QSAM).** An extended version of the basic sequential access method (BSAM).
When this method is used, a queue is formed of input data blocks that are awaiting processing or of output data
blocks that are awaiting transfer to auxiliary storage or to an output device.

2 **queued time.** See “elapsed queued time” on page 31.

quiesce. To end a process by allowing operations to complete normally, while rejecting any new requests for work.

1 **quiesce point.** A point at which data is consistent as a result of running the DB2 Universal Database QUIESCE
1 utility.

quiesced member state. In DB2 Universal Database for z/OS and OS/390, a state of a member of a data sharing group. An active member becomes quiesced when a STOP DB2 command takes effect without a failure. If the member task, address space, or DB2 Universal Database for z/OS and OS/390 system fails before the command takes effect, the member state is failed.

quoted name. See “delimited identifier” on page 26.

QSAM. See “queued sequential access method ” on page 66.

R

| **RACF.** See “Resource Access Control Facility” on page 71.

7 **RAMAC.** In a z/OS or OS/390 environment, the IBM family of enterprise disk storage system products.

7 **range-clustered table (RCT).** A table whose data is tightly clustered across one or more columns in the table. Each
7 record in the table has a predetermined offset from the logical start of the table, which allows rapid access to the
7 data.

RBA. See “relative byte address” on page 69.

7 **RCT.** (1) See “resource control table” on page 71. (2) See “range-clustered table.”

RDB. See “relational database” on page 69.

RDBMS. See “relational database management system” on page 69.

| **RDBMS catalog.** In the Information Catalog Center, a collection of tables that contains descriptions of SQL objects,
| such as tables, views, and indexes, maintained by an RDBMS.

| **readahead prefetching.** A method of prefetching pages by looking ahead in a scan, which results in asynchronous
retrieval of pages even though those pages are not located sequentially on disk. See also “sequential prefetch” on
page 75 and “list prefetch” on page 48.

7 **read-only.** Pertaining to data that can be read, but not modified or deleted.

| **read stability (RS).** An isolation level that locks only the rows that an application retrieves within a transaction.
| Read stability ensures that any qualifying row that is read during a transaction is not changed by other application
| processes until the transaction is completed, and that any row changed by another application process is not read
| until the change is committed by that process. Read stability allows more concurrency than repeatable read, and less
| concurrency than cursor stability. See also “cursor stability” on page 19, “repeatable read” on page 70, and
| “uncommitted read (UR)” on page 89.

| **read token.** The authorization key embedded in a READ PERMISSION DB DATALINK column value, returned as a
| simple column value or by using the scalar functions DLURLCOMPLETE or DLURLPATH. A read token is required
| for reading a file referenced in a READ PERMISSION DB DATALINK column.

| **real-time replication.** See “synchronous replication” on page 82.

| **rebind.** To create a package for an application program that was previously bound. For example, if an index is
added for a table that is accessed by a program, the package must be rebound for it to take advantage of the new
index. See also “automatic rebind” on page 6 and “bind” on page 8.

| **recapture.** In update-anywhere replication, to capture changes at a replica table and forward these changes to the
| master table or to other replica tables.

7 **receive queue.** In Q replication, a WebSphere MQ message queue that is used by a Q Apply program to receive
7 transactions that are captured by a Q Capture program.

record. The storage representation of a single row of a table or other data.

| **record identifier (RID).** A 3-byte page number followed by a 1-byte slot number that is used internally by DB2 to
| uniquely identify a record in a table. The RID contains enough information to address the page in which the record is
| stored. See also “row identifier” on page 73.

record identifier (RID) pool. In DB2 Universal Database for z/OS and OS/390, an area of main storage above the
16-MB line that is reserved for sorting record identifiers during list prefetch processing.

recording. The information from performance snapshots that can be viewed at a later time.

Glossary

| **recording activity monitor.** An object created by the Activity Monitor to record monitor data for database activities.
| The recorded data can be viewed at a later time.

| **record length.** The sum of a length of all the columns in a table, which is the length of the data as it is physically
| stored in the database. Records can be fixed or variable in length, depending on how the columns are defined. If all
| columns are fixed-length columns, the record is a fixed-length record. If one or more columns are variable-length
| columns, the record is a variable-length column.

recoverable log. A database log in which all log records are retained so that, in the event of a failure, lost data can
be recovered during forward recovery. See also “circular log” on page 12.

Recoverable Resource Manager Services (RRSAF). Recoverable Resource Manager Services attachment facility,
which is a DB2 Universal Database for z/OS and OS/390 subcomponent that uses OS/390 Transaction Management
and Recoverable Resource Manager Services to coordinate resource commitment between DB2 Universal Database for
z/OS and OS/390 and all other resource managers that also use OS/390 RRS in an OS/390 system.

| **recovery.** The process of rebuilding a database or table space that has become unusable because of hardware or
| software failure, or both. The process includes restoring a backup image and can also include rolling database logs
| forward in time. See also “forward recovery” on page 36 and “rollforward recovery” on page 72.

recovery log. See “database log” on page 20.

recovery pending. A state of the database or table space when it is restored from a backup. While the database or
table space is in this state, its data cannot be accessed.

recovery token. In DB2 Universal Database for z/OS and OS/390, an identifier for an element that is used in
recovery (for example, *NID* or *URID*).

7 **RECP.** See “recovery pending.”

REORP. See “REORG pending” on page 70.

recursion cycle. The cycle that occurs when a fullselect within a common table expression includes the name of the
common table expression in a FROM clause.

recursive common table expression. A common table expression that refers to itself in a FROM clause from the
fullselect. Recursive common table expressions are used to write recursive queries.

recursive query. A fullselect that uses a recursive common table expression.

redo. In DB2 Universal Database for z/OS and OS/390, a state of a unit of recovery that indicates that changes are
to be reapplied to the DASD media to ensure data integrity.

7 **referential constraint.** The referential integrity rule that the nonnull values of the foreign key are valid only if they
7 also appear as values of a parent key. See also “constraint” on page 16, “check constraint” on page 11, “informational
7 constraint” on page 43, and “unique constraint” on page 90.

| **referential cycle.** A set of referential constraints in which each table is a descendent of itself.

7 **referential integrity.** The state of a database in which all values of all foreign keys are valid. Maintaining referential
7 integrity requires the enforcement of a referential constraint on all operations that change the data in a table where
7 the referential constraints are defined.

referential structure. In DB2 Universal Database for z/OS and OS/390, a set of tables and relationships that
includes at least one table and, for every table in the set, all the relationships in which that table participates and all
the tables to which it is related.

7 **register.** In SQL replication, to define a DB2 Universal Database table, view, or nickname as a replication source.

7 **registration.** In SQL replication, the process of registering a DB2 Universal Database table, view, or nickname as a
7 replication source. Contrast with “subscription” on page 81. See also “replication source” on page 71.

7 **registration process.** In replication, the process of defining a replication source.

7 **registry database.** In a z/OS or OS/390 environment, a database of security information about principals, groups,
7 organizations, accounts, and security policies.

regular table space. A table space that can store any nontemporary data.

| **rejected transaction.** A transaction that contains one or more updates from replica tables that are in conflict with the
| master table.

| **related view.** A view that uses or is dependent on another object, such as the parent view or a table.

relational cube. A set of data and metadata that together define a multidimensional database. A relational cube is the portion of a multidimensional database that is stored in a relational database. See also “multidimensional database” on page 54.

| **relational database.** A database that can be perceived as a set of tables and manipulated in accordance with the
| relational model of data. Each database includes a set of system catalog tables that describe the logical and physical
| structure of the data, a configuration file containing the parameter values allocated for the database, and a recovery
| log with ongoing transactions and archivable transactions.

| **relational database management system (RDBMS).** A collection of hardware and software that organizes and
| provides access to a relational database.

| **relational database name (RDBNAM).** A unique identifier for a relational database within a network. In DB2
| Universal Database for z/OS and OS/390, this name must be the value in the LOCATION column of table
| SYSIBM.LOCATIONS in the communications database. DB2 Universal Database for z/OS and OS/390 publications
| refer to the name of another RDBMS as a LOCATION value or a location name.

relationship. In DB2 Universal Database for z/OS and OS/390, a defined connection between the rows of a table or the rows of two tables. A relationship is the internal representation of a referential constraint.

| **relationship category.** In the Information Catalog Center, a basis to define the relationship type. There are four
| relationship categories:

- | • Support
- | • Hierarchical
- | • Transformational
- | • Peer to Peer

| Each of these relationship categories has roles associated with it that define how an object can relate to other objects.
| For example, the support relationship category has object and support object roles available.

| **relationship type.** In the Information Catalog Center, a definition that defines the roles an object type can play in a
| relationship. The default relationship types are:

- | • Attachment
- | • Contact
- | • Contains
- | • Dictionary
- | • Input
- | • Output
- | • Linked
- | • Supported

| Each default relationship has a specific set of roles that object types can play. For example, the contains relationship
| type allows parent and child roles. If you added a contains relationship between two objects, one object takes on the
| parent role and the other object takes on the child role.

7 **relative byte address (RBA).** In a z/OS or OS/390 environment, the offset of a data record or control interval from
7 the beginning of the storage space that is allocated to the data set or file to which it belongs.

| **remigration.** The process of returning to a current release of DB2 Universal Database following a fallback to a
| previous release. This procedure constitutes another migration process.

Glossary

7 **remote.** In DB2 Universal Database for z/OS and OS/390, a property of any object that is maintained by a remote
7 DB2 Universal Database subsystem other than the local DB2 Universal Database subsystem on which the user or
7 application is located. A remote view, for example, is a view that is maintained by a remote DB2 Universal Database
7 subsystem. See also “local” on page 49.

remote attach request. In DB2 Universal Database for z/OS and OS/390, a request that is made by a remote location to attach to the local DB2 Universal Database subsystem. Specifically, the request that is sent is an SNA Function Management Header 5.

7 **remote catchup state.** In high availability disaster recovery (HADR), the state of the standby database after it reads
7 all of the existing local log files. The database remains in this state until it establishes a connection with the primary
7 database. See also “peer state” on page 61.

7 **remote database.** A database that is physically located on a system other than the one in use. See also “local
7 database” on page 49.

remote subsystem. In DB2 Universal Database for z/OS and OS/390, any RDBMS, except the *local subsystem*, with which the user or application can communicate. The subsystem need not be remote in any physical sense, and might even operate on the same processor under the same DB2 Universal Database for z/OS and OS/390 system.

| **remote unit of work (RUOW).** A unit of work that lets a user or application program read or update data at one
| location per unit of work. Remote unit of work supports access to one database within a unit of work. An application
| program can update several remote databases, but it can only access one database within a unit of work. See also
| “unit of work” on page 90.

reoptimization. The DB2 Universal Database for z/OS and OS/390 process of reconsidering the access path of an SQL statement at run time. During reoptimization, DB2 Universal Database for z/OS and OS/390 uses the values of host variables, parameter markers, or special registers.

REORG pending (REORP). In DB2 Universal Database for z/OS and OS/390, a condition that restricts SQL access and most utility access to an object that must be reorganized.

repeatable read (RR). An isolation level that locks all the rows in an application that are referenced within a transaction. When a program uses repeatable read protection, rows referenced by the program cannot be changed by other programs until the program ends the current transaction. See also “read stability” on page 67, “uncommitted read (UR)” on page 89, and “cursor stability” on page 19.

| **repeating group.** A situation in which an entity includes multiple attributes that are inherently the same. The
| presence of a repeating group violates the requirement of first normal form. In an entity that satisfies the requirement
| of the first normal form, each attribute is independent and unique in its meaning and its name. See also
| “normalization” on page 56.

| **replacement file.** In DB2 Data Links Manager, a file whose contents are intended to take the place of an existing
| file.

| **replica table.** In SQL replication, specifically in update-anywhere replication, a type of target table that can be
| updated locally and also receives updates from the master table through a subscription-set definition. If replication
| conflict detection is enabled, changes made to the replica table are rejected, whereas changes made to the master
| table are retained. See also “update-anywhere replication” on page 90, “master table” on page 52, and “conflict
| detection” on page 15.

replication. The process of maintaining a defined set of data in more than one location. It involves copying designated changes from one location (a source) to another (a target), and synchronizing the data in both locations.

7 **replication administrator.** (1) In Q replication, the user who is responsible for creating Q subscriptions and XML
7 publications. This user can also run the Q Capture program and the Q Apply program. (2) In SQL replication, the
7 user who is responsible for registering replication sources and creating subscription sets. This user can also run the
7 Capture program and the Apply program.

| **Replication Alert Monitor.** A program that checks the operation of the Capture, Apply, Q Capture, and Q Apply
| programs, and sends alerts to one or more users when it detects the specified alert conditions.

| **Replication Analyzer.** A program that can analyze a replication environment for setup problems, configuration
| errors, and performance issues.

- | **Replication Center.** A graphical interface that lets you define, operate, maintain and monitor the replication environment. It is part of the DB2 Administration Client tool suite.
- | **replication control table.** A table in which replication definitions or control information is stored.
- 7 **replication queue map.** In Q replication, an object that links a send queue and a receive queue. The replication queue map includes settings for how a Q Capture program processes all transactions that use the send queue and how a Q Apply program processes all transactions that use the receive queue. See also “publishing queue map” on page 64 and “queue map” on page 66.
- 7 **replication source.** (1) In SQL replication, a table, view, or nickname that is registered as a source for replication. Changes that are made to this table, view, or nickname are captured and copied to a target table that is defined in a subscription-set member. See also “subscription set” on page 81 and “subscription set member” on page 81. (2) In Q replication, a table that is a source for replication. Changes made to this type of table are captured and copied to a target table that is defined in a Q subscription or an XML publication. See also “Q subscription” on page 65 and “XML publication” on page 94.
- | **replication subscription.** See “subscription set” on page 81.
- 7 **replication target.** (1) In SQL replication, a table, view, or nickname that is a destination for changes that were replicated from a registered replication source. The Apply program applies these changes. See also “target table” on page 85. (2) In Q replication, a table or stored procedure that is a destination for changes that were replicated from a source. The Q Apply program applies these changes. See also “target table” on page 85.
- | **request commit.** In DB2 Universal Database for z/OS and OS/390, the vote that is submitted to the prepare phase if the participant has modified data and is prepared to commit or roll back.
- | **requester.** (1) The source of a request to access data at a remote server. Also, the system that requests the data. For DB2 Universal Database for z/OS and OS/390, the requester function is provided by distributed data facility to access a remote RDBMS. Depending on the level of DRDA protocol used, a requester can be described as an application requester or a database server. (2) The target of a request from a remote requester.
- reserved word.** (1) A word that is used in a source program to describe an action to be taken by the program or compiler. It must not appear in the program as a user-defined name or a system name. (2) A word that has been set aside for special use in the SQL standard.
- | **residual recovery entry (RRE).** A unit of recovery about which the database manager could be in doubt.
- resource.** In DB2 Universal Database for z/OS and OS/390, the object of a lock or claim, which could be a table space, an index space, a data partition, an index partition, or a logical partition.
- 7 **Resource Access Control Facility (RACF).** An IBM licensed program that provides access control by identifying users to the system; verifying users of the system; authorizing access to protected resources; logging detected, unauthorized attempts to enter the system; and logging detected access to protected resources.
- | **resource allocation.** In DB2 Universal Database for z/OS and OS/390, the part of plan allocation that deals specifically with database resources.
- resource control table (RCT).** In DB2 Universal Database for z/OS and OS/390 with CICS, a construct of the CICS attachment facility, created by site-provided macro parameters, that defines authorization and access attributes for transactions or transaction groups.
- 7 **resource definition online.** In a z/OS or OS/390 environment with CICS, a feature that is used to define CICS resources online without assembling tables.
- resource limit facility (RLF).** A portion of DB2 Universal Database for z/OS and OS/390 code that prevents dynamic manipulative SQL statements from exceeding specified time limits. Also known as the *governor*.
- resource limit specification table.** In DB2 Universal Database for z/OS and OS/390, a site-defined table that specifies the limits to be enforced by the resource limit facility.
- | **response file.** An ASCII file that can be customized with the setup and configuration data that will automate an installation. The setup and configuration data must be entered during an interactive install, but with a response file, the installation can proceed without any intervention.

Glossary

- | **response file generator.** A utility that creates a response file from an existing installed and configured DB2 Universal Database product. You can use the generated response file to recreate the same setup on other computers.
- restart pending (RESTP).** In DB2 Universal Database for z/OS and OS/390, a restrictive state of a page set or partition that indicates that restart (backout) work needs to be performed on the object. All access to the page set or partition is denied except for access by the RECOVER POSTPONED command or the automatic online backout, which DB2 Universal Database for z/OS and OS/390 invokes after restart if the system parameter LBACKOUT=AUTO.
- RESTP.** See “restart pending.”
- | **restore.** To rebuild a damaged or corrupted database or table space from a backup image produced with the BACKUP utility.
- | **restore set.** A backup copy of a database or table space plus zero or more log files, which, when restored and rolled forward, bring the database or table space back to a consistent state.
- 2 **result destination.** In Query Patroller, where the results of a query are returned. The result destination can be either
2 the application through which the query was submitted or a result table. See also “result table.”
- result set.** The set of rows that a stored procedure returns.
- result set locator.** A 4-byte value that DB2 Universal Database for z/OS and OS/390 uses to uniquely identify a query result set that a stored procedure returns.
- result table.** (1) The set of rows produced by the evaluation of a SELECT statement. See also “temporary table” on
2 page 85. (2) In Query Patroller, a table created to store the result set of a query.
- | **retained lock.** A MODIFY lock that a DB2 Universal Database for z/OS and OS/390 subsystem was holding at the
| time of a subsystem failure. The lock is retained in the coupling facility lock structure across a DB2 Universal
| Database for z/OS and OS/390 subsystem.
- | **retention-limit pruning.** In SQL replication, the pruning of CD and unit-of-work tables by the Capture program that
| are older than a limit that the user specifies.
- | **revoke.** To remove a privilege or authority from an authorization identifier.
- | **rework.** (1) To convert an insert into a replication target table to an update if the insert fails because the row already
| exists in the target table. (2) To convert an update to a replication target table to an insert if the update fails because
| the row does not exist in the target table.
- RID.** See “record identifier” on page 67.
- RID pool.** See “record identifier pool” on page 67.
- right outer join.** The result of a join operation that includes the matched rows of both tables that are being joined and preserves the unmatched rows of the second join operand. See “join” on page 46. See also “left outer join” on page 48 and “full outer join” on page 36.
- RLF.** See “resource limit facility” on page 71.
- | **role.** In the Information Catalog Center, a descriptor that is associated with the relationship category. The
| relationship category determines what roles are available for each object type.
- | **roll forward.** To update the data in a restored database or table space by applying changes recorded in the database
7 log files. See also “rollforward recovery.”
- | **rollforward recovery.** A process that is started through the rollforward utility that is used to recover a database by
7 applying transactions that were recorded in the database recovery log file. See also “version recovery” on page 92.
- | **roll back.** To restore data that is changed by an SQL statements to the state at its last commit point. See “backout”
| on page 7. See also “point of consistency” on page 62.

| **roll out.** The efficient deletion of a large portion of a multidimensional clustering (MDC) table, and is possible when
 | a DELETE statement is processed that has certain types of predicates (equality, range, BETWEEN, IN) on one or more
 | dimension columns. Most logging can be avoided and, in certain cases, all per-row processing can also be avoided.

root page. In DB2 Universal Database for z/OS and OS/390, the page of an index page set that follows the first
 index space map page. A root page is the highest level (or the beginning point) of the index.

7 **routine.** A database object that encapsulates procedural logic and SQL statements, is stored on the database server,
 7 and can be invoked from an SQL statement or by using the CALL statement. The three main classes of routines are
 7 procedures, functions, and methods. See also “stored procedure” on page 80, “function” on page 36, and “method”
 7 on page 53.

row. The horizontal component of a table that consists of a sequence of values, one for each column of the table.

| **row-capture rules.** In SQL replication, rules based on changes to registered columns that define when and whether
 | the Capture program writes a row to a CD table, or when and whether the Capture triggers write a row to a CCD
 | table.

| **row function.** An SQL function that optionally accepts arguments and that returns a single row of values. A row
 | function can be implemented in SQL and used as a transform function to map attributes of a structured type to
 | built-in data type values in a row. See also “function” on page 36, “aggregate function” on page 2, “scalar function”
 | on page 74, and “table function” on page 84.

ROWID. See “row identifier.”

| **row identifier (ROWID).** A value that uniquely identifies a row. This value is stored with the row and does not
 | change.

| **row lock.** A lock on a single row of data. See also “lock” on page 49 and “table lock” on page 84.

7 **row-positioned access.** The ability to retrieve a single row from the single FETCH statement.

7 **rowset.** A set of rows for which a cursor position is established.

7 **rowset cursor.** A cursor that is defined so that one or more rows can be returned as a rowset for a single FETCH
 7 statement, and the cursor is positioned on the set of rows that is fetched.

7 **rowset-positioned access.** The ability to retrieve multiple rows from a single FETCH statement.

7 **row trigger.** In DB2 Universal Database for z/OS and OS/390, a trigger whose granularity is defined by using the
 7 FOR EACH ROW clause.

RR. See “repeatable read” on page 70.

| **RRE.** See “residual recovery entry” on page 71.

RS. See “read stability” on page 67.

RRSAF. See “Recoverable Resource Manager Services” on page 68.

RUOW. See “remote unit of work” on page 70.

S

7 **satellite.** A DB2 Universal Database server that is a member of a group of similar DB2 Universal Database servers.
 7 Each satellite in the group runs the same application, and has a similar configuration to support the application.

Satellite Administration Center. A user interface that provides centralized administrative support for satellites.

7 **satellite control server.** A DB2 Universal Database system that contains the satellite control database, SATCTLDB.

| **saved search.** In the Information Catalog Center, a set of search criteria that is saved for subsequent use. A saved
 | search is displayed as an object under the **Saved Searches** folder in the tree.

Glossary

- 7 **savepoint.** A named entity that represents the state of data and schemas at a particular point in time within a unit
7 of work.
- 7 **savepoint level.** A distinct scope that is used for reference and for interaction between savepoint-related statements.
- SBCS.** See “single-byte character set” on page 77.
- 7 **SCA.** See “shared communications area” on page 76.
- scalar fullselect.** A fullselect that returns a single value—one row of data that consists of exactly one column.
- 7 **scalar function.** A function that optionally accepts arguments and that returns a single scalar value each time that it
7 is invoked. A scalar function can be referenced in an SQL statement wherever an expression is valid. See also
7 “function” on page 36, “aggregate function” on page 2, “row function” on page 73, and “table function” on page 84.
- 7 **scalar method.** A method that optionally accepts arguments and that returns a single scalar value each time that it is
7 invoked. See also “method” on page 53.
- scale.** The number of digits in the fractional part of a number.
- l **scattered read.** A method of reading contiguous data pages from disk to discontinuous portions of memory. See also
l “block based I/O” on page 8.
- l **schema.** (1) A collection of database objects such as tables, views, indexes, or triggers that define a database. A
l database schema provides a logical classification of database objects. (2) In DB2 Universal Database for z/OS and
OS/390, a logical grouping for user-defined functions, distinct types, triggers, and stored procedures. When an object
of one of these types is created, it is assigned to one schema, based on the name of the object. (3) In the Data
Warehouse Center, a collection of warehouse target tables and the relationships between the warehouse target table
columns, where the target tables can come from one or more warehouse targets.
- l **scrollability.** In a z/OS or OS/390 environment, the ability to use a cursor to fetch in either a forward or a
l backward direction. The FETCH statement supports multiple fetch orientations to indicate the new position of the
l cursor. See also “fetch orientation” on page 35.
- l **scrollable cursor.** A cursor that can be moved in both a forward and a backward direction. See also “nonscrollable
l cursor” on page 56.
- 7 **scrollable result set.** A result set that is associated with a scrollable cursor that allows the application to fetch rows
7 and to refetch previously fetched rows. See also “result set” on page 72.
- 7 **SDWA.** See “system diagnostic work area” on page 83.
- l **search.** To request the display of objects that meet user-specified criteria.
- search condition.** A criterion for selecting rows from a table. A search condition consists of one or more predicates.
- l **search criteria.** In the Information Catalog Center, options and character strings that are used to specify how to
l perform a search. The search criteria can include object type names, property values, whether the search is for an
l exact match, and whether the search is case sensitive.
- l **secondary authorization ID.** In DB2 Universal Database for z/OS and OS/390, an authorization identifier that is
l associated with a primary authorization ID by an authorization exit routine.
- secondary group buffer pool.** For a duplexed group buffer pool in a DB2 Universal Database for z/OS and OS/390
environment, the structure that is used to back up changed pages that are written to the primary group buffer pool.
No page registration or cross-invalidation occurs using the secondary group buffer pool. The z/OS and OS/390
equivalent is *new* structure. See also “primary group buffer pool” on page 63.
- 7 **secondary index.** An index that is defined on a partitioned table space and does not meet the definition of the
7 partitioned index.
- secondary log.** A set of one or more log files used to record changes to a database. Storage for these files is allocated
as needed when the primary log is full. See also “primary log” on page 63.

section. The segment of a plan or package that contains the executable structures for a single SQL statement. For most SQL statements, one section in the plan exists for each SQL statement in the source program. However, for cursor-related statements, the DECLARE, OPEN, FETCH, and CLOSE statements reference the same section because they each refer to the SELECT statement that is named in the DECLARE CURSOR statement. SQL statements such as COMMIT, ROLLBACK, and some SET statements do not use a section.

| **segment.** A group of pages that hold a row of a single table. See also “segmented table space.”

segmented table space. In DB2 Universal Database for z/OS and OS/390, 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.

| **self-referencing constraint.** A referential constraint that defines a relationship in which a table is a dependent of itself.

self-referencing row. A row that is a parent of itself.

self-referencing subquery. A subselect or fullselect within a DELETE, INSERT, or UPDATE statement that refers to the same table that is the object of the SQL statement.

self-referencing table. A table that is both a parent and a dependent table in the same referential constraint.

send queue. In Q replication, a WebSphere MQ message queue that is used by a Q Capture program to publish transactions that it has captured. A send queue can be used either for Q replication or event publishing, but not both at the same time.

| **sensitive cursor.** A cursor that is sensitive to changes made to the database after the result table has materialized. See also “insensitive cursor” on page 44.

7 **sensitivity.** The amount of time by which a threshold-based health indicator must exceed its threshold, or the
7 amount of time that a state-based health indicator must be in a non-normal state before an alert is generated.

7 **sequence.** A database object that is independent of any one table that automatically generates unique key values
7 based on initial user specifications.

sequential data set. A non-DB2 Universal Database for z/OS and OS/390 data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Several of the DB2 Universal Database for z/OS and OS/390 database utilities require sequential data sets.

| **sequential prefetch.** A mechanism that triggers consecutive asynchronous I/O operations. Pages are fetched before
| they are required, and several pages are read with a single I/O operation.

| **serial cursor.** See “nonscrollable cursor” on page 56.

| **serialization.** (1) The consecutive ordering of items. (2) The process of controlling access to a resource to protect the
| integrity of the resource. (3) In Q replication, the process of applying transactions in the same order that they were
| committed at the source.

7 **server.** (1) In a network, hardware or software that provides facilities to other stations, for example, a file server, a
7 printer server, or a mail server. (2) The target of a request from a remote requester. In a DB2 Universal Database
7 system, the server function is provided by the distributed data facility, which is used to access DB2 Universal
7 Database data from remote applications. See also “application server” on page 4. (3) See “logical server” on page 50.
7 See also “Apply control server” on page 4, “Apply server” on page 4, “Capture control server” on page 10, “control
7 server” on page 17, “Monitor control server” on page 53, “Q Apply server” on page 65, “Q Capture server” on page
7 65, “source server” on page 77, and “target server” on page 85.

| **server definition.** In a federated system, the name and information that define the data sources to the federated
| database. The server definition is used by the wrapper when SQL statements that use nicknames are submitted to the
| federated database.

| **server option.** In a federated system, information within a server definition that either configures the wrapper itself,
| or affects the way that DB2 Universal Database uses the wrapper. Server option values are stored in the global
| catalog.

Glossary

- | **server profile.** A profile that contains information about server instances on a system, and databases within each server instance. See also “client profile” on page 12.
- | **server-side programming.** A method for adding DB2 Universal Database data into dynamic Web pages.
service class. In DB2 Universal Database for z/OS and OS/390, an 8-character identifier that is used by MVS Workload Manager to associate customer performance goals with a particular DDF thread or stored procedure. A service class is also used to classify work on parallelism assistants.
- | **service definition.** In a federated database system, a description of a data source.
service name. A name that provides a symbolic method of specifying the port number to be used at a remote node. The TCP/IP connection requires the address of the remote node and the port number to be used on the remote node to identify an application.
session. A logical connection between two stations or SNA network addressable units (NAUs) that allows the two stations or NAUs to communicate.
session limit. In SNA, the maximum number of concurrently active logical unit to logical unit (LU-to-LU) sessions that a particular logical unit (LU) can support.
session partner. In SNA, one of the two network addressable units (NAUs) participating in an active session.
session protocols. In DB2 Universal Database for z/OS and OS/390, the available set of SNA communication requests and responses.
session security. For LU 6.2, partner LU verification and session data encryption. A Systems Network Architecture (SNA) function that allows data to be transmitted in encrypted form.
set operator. The SQL operators UNION, EXCEPT, and INTERSECT that correspond to the relational operators union, difference, and intersection. A set operator derives a result table by combining two other result tables.
- | **shadow index.** A new index structure created during the index reorganization. It is not visible to users for access until the database manager has fully rebuilt the index.
- | **shadowing.** A recovery technique in which current page contents are never overwritten. Instead, new pages are allocated and written while the pages whose values are being replaced are retained as shadow copies until they are no longer needed to support the restoration of the system state due to a transaction rollback.
shared communications area (SCA). A coupling facility list structure that a DB2 Universal Database for z/OS and OS/390 data sharing group uses for inter-DB2 communication.
- | **shared lock.** A lock that limits concurrently executing application processes to read-only operations on database data. See also “exclusive lock” on page 32.
shift-in character. A special control character (X'0F') that is used in EBCDIC systems to denote that the subsequent bytes represent SBCS characters. See also “shift-out character.”
shift-out character. A special control character (X'0E') that is used in EBCDIC systems to denote that the subsequent bytes, up to the next shift-in control character, represent DBCS characters. See also “shift-in character.”
short string. (1) A fixed-length string or a variable-length string whose maximum length is less than or equal to 254 bytes. (2) In DB2 Universal Database for z/OS and OS/390, a string whose actual length, or a variable-length string whose maximum length, is 255 bytes (or 127 double-byte characters) or less. Regardless of length, a LOB string is not a short string. See also “long string” on page 51.
- | **signal.** In replication, an SQL statement that allows communication with the Capture program and the Q Capture program. A signal is inserted into the signal control table and received by the Capture program or the Q Capture program when the program reads the log entry for the signal insert.
- | **sign on.** A request that is made on behalf of an individual CICS or IMS application process by an attachment facility to enable DB2 Universal Database for z/OS and OS/390 to verify that it is authorized to use DB2 Universal Database resources.

7 **Simple Object Access Protocol (SOAP).** An XML-based protocol for the exchange of information in a decentralized,
7 distributed environment.

simple page set. In DB2 Universal Database for z/OS and OS/390, a nonpartitioned page set. A simple page set initially consists of a single data set (page set piece). If that data set is extended to 2 gigabytes, another data set is created, and so on up to a total of 32 data sets. DB2 Universal Database for z/OS and OS/390 considers the data sets to be a single contiguous linear address space that contains a maximum of 64 gigabytes. Data is stored in the next available location within this address space without regard to any partitioning scheme.

simple table space. In DB2 Universal Database for z/OS and OS/390, a table space that is neither partitioned nor segmented.

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. See also “double-byte character set” on page 30 and “multibyte character set” on page 54.

single-precision floating point number. A 32-bit approximate representation of a real number.

| **slice.** The set of blocks that contain pages with data having a certain value of one of the clustering dimensions. If
| we consider a slice in each dimension, where a slice contains a particular value for its dimension, a cell is the
| intersection of these slices.

SMF. See “system management facility” on page 83.

SMS. See “Storage Management Subsystem” on page 80.

SMS table space. See “system-managed space table space ” on page 83.

SNA. See “Systems Network Architecture” on page 83.

SNA network. The part of the user application network that conforms to the formats and protocols of Systems Network Architecture (SNA). It enables reliable transfer of data among users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), gateway function, intermediate session routing function components, and the transport network.

| **snapshot.** A record of the current state of the database environment. See also “performance snapshot” on page 61,
| “explain” on page 33, and “health snapshot” on page 39.

7 **SOAP.** See “Simple Object Access Protocol.”

| **socket.** A communications handle used by TCP/IP.

| **socket interface.** A callable TCP/IP programming interface that is used by TCP/IP network applications to communicate with remote TCP/IP partners.

soft checkpoint. The process of writing some information to the log file header; this information is used to determine the starting point in the log if a database restart is required.

| **source.** In the Data Warehouse Center, a table, view, or file that is input to a step. See also “target” on page 84.

7 **sourced function.** A function that duplicates the semantics of another function, called a source function. Only scalar
7 and aggregate functions can be sourced functions. See also “external function” on page 34, “user-defined function”
7 on page 91, “built-in function” on page 9, and “SQL function” on page 78.

source program. A set of host language statements and SQL statements that is processed by an SQL precompiler.

7 **source server.** A database or subsystem that contains the source tables for replication.

7 **source table.** (1) A table that can be a base table, a view, a table expression, or a user-defined table function. (2) A
| table that contains data that is to be replicated to a target table. Contrast with “target table” on page 85.

source type. An existing type that is used to internally represent a distinct type.

spatial column. A table column or view column that has a spatial data type. This data type allows the column to contain coordinates that define locations within a particular region of the earth.

Glossary

7 **spatial reference system.** In DB2 Spatial Extender and DB2 Geodetic Extender, a set of parameters that includes
7 coordinates that define the maximum possible extent of space that is referenced by a given range of coordinates, an
7 identifier of the coordinate system from which the coordinates are derived, and numbers that convert coordinates
7 into positive integers to improve performance when the coordinates are processed.

special register. A storage area that is defined for an application process by the database manager and is used to store information that can be referenced in SQL statements. Examples are USER and CURRENT DATE.

7 **specific function name.** (1) The name that uniquely identifies a function to the system. (2) In DB2 Universal
Database for z/OS and OS/390, a particular user-defined function that is known to the database manager by its
specific name. When a user-defined function is defined to the database, every function is assigned a specific name
that is unique within its schema. The specific name is important for functions that have the same name but have
either a different number of parameters or different data types associated with those parameters. The user can either
1 provide this name or use the default.

7 **spill agent thread.** In Q replication, a thread that applies transactions that are waiting in the spill queue and
7 informs the browser thread when the spill queue is empty and deleted.

7 **spill file.** In SQL replication, a temporary file that the Apply program creates to hold data for updating target
7 tables.

7 **spill queue.** In Q replication, a dynamic queue that the Q Apply program creates to hold transactions that occur at
7 the source table while a target table is being loaded. The Q Apply program later applies these transactions and then
7 deletes the spill queue.

SPUFI. See "SQL Processor Using File Input" on page 79.

SQL. See "Structured Query Language" on page 81.

1 **SQL Assistant.** A wizard that is available in several DB2 Universal Database tools and centers that generates SQL
1 statements graphically.

SQL authorization ID (SQL ID). In DB2 Universal Database for z/OS and OS/390, the authorization identifier that is used for checking dynamic SQL statements in some situations.

SQLCA. See "SQL communication area."

SQL communication area (SQLCA). A set of variables that provides an application program with information about the execution of its SQL statements or its requests from the database manager.

SQL connection. An association between an application process and a local or remote application server.

SQLDA. See "SQL descriptor area."

7 **SQL data change statement.** An SQL statement that is used to make changes to data that is stored in user tables,
7 including the INSERT, UPDATE, DELETE, and MERGE statements.

SQL descriptor area (SQLDA). (1) A set of variables that is used in the processing of certain SQL statements. The SQLDA is intended for dynamic SQL programs. (2) A structure that describes input variables, output variables, or the columns of a result table.

1 **SQL escape character.** The symbol that is used to enclose an SQL delimited identifier. The escape character is the
1 quotation mark, except in COBOL applications, where the user assigns the symbol to be either a quotation mark or
1 an apostrophe.

1 **SQL function.** A function that is implemented entirely by using a subset of SQL statements and SQL PL statements.
See also "function" on page 36, "built-in function" on page 9, "sourced function" on page 77, and "external function"
1 on page 34.

SQL ID. See "SQL authorization ID."

1 **SQLJ.** A three-part standard for supporting embedded SQL in Java programs (Part 0), defining and calling Java
1 stored procedures and user-defined functions (Part 1), and using database structured types in Java (Part 2).

7 **SQL method.** A method that is implemented entirely by using a subset of SQL statements and SQL PL statements.
7 See also “method” on page 53 and “external method” on page 34.

l **SQL path.** In DB2 Universal Database for z/OS and OS/390, an ordered list of schema names that is used in the
l resolution of unqualified references to user-defined functions, distinct types, and stored procedures. In dynamic SQL,
l the current path is found in the CURRENT PATH special register. In static SQL, it is defined in the PATH bind
l option.

7 **SQL PL.** See “SQL procedural language.”

7 **SQL procedural language (SQL PL).** A language extension of SQL that consists of statements and language
7 elements that can be used to implement procedural logic in SQL statements. SQL PL provides statements for
7 declaring variables and condition handlers, assigning values to variables, and for implementing procedural logic.

l **SQL procedure.** A procedure that is created by running the CREATE PROCEDURE statement and is implemented
l entirely in SQL PL. An SQL procedure is called by running the CALL statement. See also “procedure” on page 64
l and “external procedure” on page 34.

SQL processing conversation. Any conversation that requires access to DB2 Universal Database for z/OS and OS/390 data, either through an application or by dynamic query requests.

l **SQL Processor Using File Input (SPUFI).** In DB2 Universal Database for z/OS and OS/390, a facility of the TSO
l attachment subcomponent that lets the DB2I user run SQL statements without embedding them in an application
l program.

7 **SQL replication.** A type of replication that uses staging tables.

SQL return code. Either SQLCODE or SQLSTATE.

SQL routine. In DB2 Universal Database for z/OS and OS/390, a user-defined function or stored procedure that is based on code that is written in SQL.

l **SQL statement coprocessor.** In a z/OS or OS/390 environment, an alternative to the DB2 Universal Database
l precompiler that lets the user process SQL statements at compile time. The user invokes an SQL statement
l coprocessor by specifying a compiler option.

SQL string delimiter. In DB2 Universal Database for z/OS and OS/390, a symbol that is used to enclose an SQL string constant. The SQL string delimiter is the apostrophe ('), except in COBOL applications, where the user assigns the symbol to be either an apostrophe or a quotation mark (").

SSCP. See “system services control point” on page 83.

7 **SSI.** See “subsystem interface” on page 82.

SSM. In DB2 Universal Database for z/OS and OS/390, subsystem member.

stack. An area in memory that stores temporary register information, parameters, and return addresses of subroutines.

staging table. In SQL replication, a CCD table that is used to save data before that data is replicated to the target database. A CCD table that is used for staging data can function as an intermediate source for updating data to one or more target tables. See also “consistent-change-data table” on page 16.

standalone. An attribute of a program that allows the program to run separately from DB2 Universal Database for z/OS and OS/390, without using DB2 Universal Database for z/OS and OS/390 services.

7 **standard database.** In high availability disaster recovery (HADR), a database that is neither the primary nor the
7 standby. A standard database is not configured for HADR.

7 **standby database.** In high availability disaster recovery (HADR), a copy of the primary database. Updates to this
7 database occur by rolling forward log data that is generated on the primary database and sent to the standby
7 database.

Glossary

star schema. A type of relational database schema that is composed of a set of tables comprised of a single, central fact table surrounded by dimension tables. Star schemas are used by the DB2 OLAP Server and are often created in the Data Warehouse Center.

statement. An instruction in a program or procedure.

7 **statement handle.** In the CLI, a handle that refers to the data object that contains information about an SQL
1 statement. Such information includes dynamic arguments, bindings for dynamic arguments and columns, cursor
1 information, result values, and status information. Each statement handle is associated with a connection handle.

7 **statement savepoint.** An internal savepoint mechanism that ensures that, at the completion of an SQL statement,
7 either all updates are applied to the database or none of the updates are applied to the database. Compare with
7 “commit” on page 14.

statement string. For a dynamic SQL statement in a DB2 Universal Database for z/OS and OS/390 environment, the character string form of the statement.

7 **statement trigger.** In DB2 Universal Database for z/OS and OS/390, a trigger whose granularity is defined by using
7 the FOR EACH STATEMENT clause. See also “trigger” on page 88.

1 **static bind.** A process by which SQL statements are bound after they are precompiled. All static SQL statements are
1 prepared for execution at the same time. See “bind” on page 8. See also “dynamic bind” on page 30.

7 **static cursor.** A named control structure that does not change the size of the result table or the order of its rows after
7 an application opens the cursor. See also “dynamic cursor” on page 30.

1 **static SQL.** SQL statements that are embedded within a program, and are prepared during the program preparation
1 process before the program is executed. After being prepared, a static SQL statement does not change, although
1 values of host variables specified by the statement can change. See also “embedded SQL” on page 31 and “dynamic
1 SQL” on page 30.

7 **statistics profile.** A file that contains all of the options information that specifies which statistics are collected for a
7 table when using a particular RUNSTATS command logic.

status. In the Data Warehouse Center, the work-in-progress processing condition of a step, such as scheduled, populating, or successful.

1 **step.** In the Data Warehouse Center, a single operation on data in a warehouse process. In most cases, a step
1 includes a warehouse source, a description of the transformation or movement of data, and a target. A step can be
1 run according to a schedule, or it can cascade from another step.

step edition. In the Data Warehouse Center, a snapshot of the data in a warehouse source at a particular time.

1 **storage group.** A named set of disks on which DB2 Universal Database for z/OS and OS/390 data can be stored.

1 **Storage Management Subsystem (SMS).** In OS/390, software that automates as much as possible the management
1 of physical storage by centralizing control, automating tasks, and providing interactive controls for system
1 administrators. SMS can reduce users’ concerns about physical details of performance, space, and device
1 management.

1 **stored procedure.** (1) An application program, possibly containing SQL statements, that can be invoked with the
1 SQL CALL statement. (2) A user-written application program that can be started through the use of the SQL CALL
1 statement.

Stored Procedure Builder. Renamed and enhanced in DB2 Universal Database Version 8. See “Development Center” on page 27.

storyboard. A visual summary of a video. The Video Extender includes features that can be used to identify and store video frames that are representative of the shots in a video. These representative frames can be used to build a storyboard.

1 **string.** (1) In programming languages, the form of data that is used for storing and manipulating text. (2) A
1 sequence of bytes that might represent characters.

7 **strong typing.** A process that guarantees that only user-defined functions and operations that are defined on a
 7 distinct type can be applied to that type. For example, two currency types, such as Canadian dollars and US dollars,
 7 cannot be directly compared, but a user-defined function can be provided to convert one currency to the other and
 7 then do the comparison.

| **structure.** A name that refers collectively to different types of DB2 Universal Database objects, such as tables,
 | databases, views, indexes, and table spaces.

Structured Query Language (SQL). A standardized language for defining and manipulating data in a relational
 database.

| **structured type.** A data type that is a named collection of attributes (standard data types or other structured types),
 | which allows for greater semantic control than predefined types.

| **subagent.** A type of agent that works on subrequests. A single application can make many requests, and each
 | request can be broken into many subrequests. Therefore, there can be multiple subagents that work on behalf of the
 | same application. All subagents working for the application are initiated by the initiating agent for that application.
 | See also “coordinating agent” on page 17.

subcomponent. A group of closely related DB2 Universal Database for z/OS and OS/390 modules that work
 together to provide a general function.

subject area. (1) In the Data Warehouse Center, a set of processes that create warehouse data for a particular logical
 business area. Processes in a subject area operate on data for a particular subject to create the detail data, data
 | summaries, and cubes needed by that subject. (2) In the Information Catalog Center, an object type that identifies and
 | groups the processes that relate to a logical area of the business. For example, if you are creating an information
 | catalog of marketing and sales data, you define object types Sales and Marketing and select to make them subject
 | areas. Then any objects of type Sales or Marketing are grouped under the corresponding subject.

| **subject search.** See “browse” on page 8.

| **subject table.** The table for which a trigger is created. When the defined triggering event occurs on this table, the
 trigger is activated.

2 **submitter.** In the Query Patroller environment, a user given access to submit queries.

subordinate agent. See “subagent.”

subpage. In DB2 Universal Database for z/OS and OS/390, the unit into which a physical index page can be
 divided.

subquery. A SELECT statement within the WHERE or HAVING clause of another SQL statement; a nested SQL
 statement.

| **subscription.** (1) In SQL replication, an object that creates subscription sets and subscription-set members. Contrast
 | with “registration” on page 68 in SQL replication and “Q subscription” on page 65 in Q replication. (2) See also
 | “subscription set.”

| **subscription cycle.** The process in which the Apply program retrieves changed data for a given subscription set,
 | replicates the changes to the target table, and updates the appropriate replication control tables to reflect its status
 | and current progress.

7 **subscription set.** In SQL replication, a definition that controls the replication of changed data during a subscription
 7 cycle. A subscription set can contain zero or more subscription-set members.

7 **subscription-set member.** In SQL replication, a definition that maps a registered replication source with a replication
 7 target. Each member defines the structure of the target table and the rows and columns that will be replicated from
 7 the source table.

subselect. The form of a query that does not include an ORDER BY clause, an UPDATE clause, or UNION
 operators.

| **subset.** To replicate data from part of a source table, rather than from the entire table, to a target table. Data can be
 | subset by rows or by columns.

Glossary

substitution character. In SQL, a unique character that is substituted during character conversion for any characters in the source program that do not have a match in the target coding representation.

subsystem. In DB2 Universal Database for z/OS and OS/390, a distinct instance of a relational database management system (RDBMS).

7 **subsystem interface.** The MVS interface by which routines (IBM-supplied, vendor-written, or installation-written)
7 request services of, or pass information to, subsystems. The subsystem interface is used by Tivoli NetView for
7 OS/390 to receive system messages and enter system commands (when used with extended MCS consoles, it is used
7 to receive commands, not messages), and to communicate with other instances of Tivoli NetView for OS/390.

| **success code set.** One or more expressions that specify the return codes of a successful task. For example, specify >
| -1 to consider all return codes of zero or more a success.

| **summary table.** A specialized type of materialized query table whose fullselect contains a GROUP BY clause which
| summarizes data from the tables referenced in the fullselect. See also “materialized query table” on page 52.

| **superuser.** A user that has various system control authorities above and beyond that of the ordinary user. In UNIX
| environments, the standard superuser is root.

| **support relationship category.** In the Information Catalog Center, a category for relationship types that connects
| supporting objects to another object. For example, you can connect a News object to a Spreadsheet object.

| **support relationship type.** In the Information Catalog Center, a category of relationship types that provides
| additional information about your information catalog or enterprise. For example, the “Information Catalog Center
| News” object type in the sample information catalog. See also “relationship type” on page 69.

| **surrogate pair.** A coded representation for a single character that consists of a sequence of two Unicode values,
| where the first value of the pair is a high-surrogate in the range U+D800 through U+DBFF, and the second value is a
| low-surrogate in the range U+DC00 through U+DFFF. Surrogate pairs provide an extension mechanism for encoding
| 917 476 characters without requiring the use of 32-bit characters.

symbolic destination name. The name of a remote partner. The name corresponds to an entry in the CPI-C side information table that contains the necessary information (partner LU name, mode name, partner TP name) for the client to set up an APPC connection to the server.

| **synchpoint.** In SQL replication, a control table value for the DB2 Universal Database log or journal record sequence
| number of the last change that is applied during the most recent Apply cycle. This value is also used to coordinate
| the pruning of CD tables.

7 **synchronization.** The process by which a satellite downloads and runs the same DB2 Universal Database
7 commands, operating system commands, and SQL statements from the satellite control server as the other members
7 of its group, and then reports the results to the satellite control server.

synchronization level. In APPC, the specification indicating whether the corresponding transaction programs exchange confirmation requests and replies.

synchronous. Pertaining to two or more processes that depend on the occurrences of specific events, such as a common timing signal. See also “asynchronous” on page 5.

7 **synchronous mode.** In high availability disaster recovery (HADR), the synchronization mode in which the primary
7 database considers a transaction to be committed when it gets an acknowledgement message from the standby
7 system that confirms that the relevant log data was received and written to disk on the standby system.

| **synchronous replication.** Also known as real-time replication, a type of replication that delivers updates
| continuously and within the scope of source transactions.

sync point. See “point of consistency” on page 62.

synonym. In DB2 Universal Database for z/OS and OS/390, an alternative name, in SQL, for a table or view.

- syntactic character set.** A set of 81 graphic characters that are registered in the IBM registry as character set 00640. This set is used for syntactic purposes maximizing portability and interchangeability across systems and country or region boundaries. It is contained in most of the primary registered character sets, with a few exceptions. See also “invariant character set” on page 46.
- Sysplex.** See “Parallel Sysplex” on page 59.
- Sysplex query parallelism.** Parallel execution of a single query that is accomplished by using multiple tasks on more than one DB2 Universal Database for z/OS and OS/390 subsystem. See also “query CP parallelism” on page 66.
- system administrator.** (1) The person at a computer installation who designs, controls, and manages the use of the computer system. (2) A DB2 Universal Database user with SYSADM authority.
- system agent.** A work request that DB2 Universal Database for z/OS and OS/390 creates internally, such as prefetch processing, deferred writes, and service tasks. See also “agent” on page 2.
- system authority.** SYSCTRL and SYSMANT authority levels with full privileges for managing the system but without the ability to access the data.
- system catalog.** See “catalog” on page 10.
- system conversation.** The conversation that two DB2 Universal Database for z/OS and OS/390 subsystems must establish to process system messages before any distributed processing can begin.
- system database directory.** A directory that contains entries for every database that can be accessed using the database manager. The directory is created when the first database is created or cataloged on the system. See also “local database directory” on page 49.
- system diagnostic work area (SDWA).** In a z/OS or OS/390 environment, the data that is recorded in a SYS1.LOGREC entry that describes a program or hardware error.
- system-directed connection.** A connection that an RDBMS manages by processing SQL statements with three-part names (or nicknames), providing a level of location transparency. See also “application-directed connections” on page 4.
- system management facility (SMF).** In DB2 Universal Database for z/OS and OS/390, a standard feature that collects and records a variety of system and job-related information. For example, statistics, accounting information, and performance data.
- system-managed space (SMS) table space.** A table space whose space is managed by the operating system. This storage model is based on files that are created under subdirectories and managed by the file system. See also “database-managed space table space” on page 20.
- system monitor.** See “database system monitor” on page 21.
- system services control point (SSCP).** The control point in an SNA network that provides network services for dependent nodes.
- system time.** In UNIX, represents the time spent in system calls. See also “user time” on page 92.
- Systems Network Architecture (SNA).** An architecture that describes the logical structure, formats, protocols, and operational sequences for transmitting information units through networks, and also operational sequences for controlling the configuration and operation of networks.
- SYS1.LOGREC.** In a z/OS or OS/390 environment, a service aid that contains information about program and hardware errors.

T

- table.** A named data object consisting of a specific number of columns and some unordered rows. See also “base table” on page 7, “declared temporary table” on page 26, and “temporary table” on page 85.
- table check constraint.** See “check constraint” on page 11.

Glossary

1 **table collocation.** In a partitioned database environment, a state that occurs when two tables are stored in the same
1 database partition group and that have the same number of compatible partitioning keys. When this happens, DB2
1 Universal Database can choose to perform the join or subquery processing at the database partition where the data is
1 stored.

7 **table-controlled partitioning.** A type of partitioning in which partition boundaries for a partitioned table are
7 controlled by values that are defined in the CREATE TABLE statement.

table designator. A column name qualifier that designates a specific object table.

1 **table expression.** An expression that creates a temporary result table from a simple query. For example, a table
1 expression might be a query that selects all the managers from several departments and further specifies that they
1 have over 15 years of working experience and are located at the main branch. See also “common table expression” on
1 page 14.

7 **table function.** A function that optionally accepts arguments and that returns a table to the SQL statement that
7 references it. Table functions can be referenced only in the FROM clause. See also “function” on page 36, “aggregate
7 function” on page 2, “scalar function” on page 74, and “row function” on page 73.

table locator. In DB2 Universal Database for z/OS and OS/390, a mechanism that allows access to trigger transition
tables in the FROM clause of SELECT statements, the subselect of INSERT statements, or from within user-defined
functions. A table locator is a fullword integer value that represents a transition table.

1 **table lock.** A lock on a table of data. See also “row lock” on page 73 and “row identifier” on page 73.

1 **table-mode processing.** In SQL replication, a type of replication subscription-set processing in which the Apply
1 program retrieves all the data from the source CD table, then applies the data (one member at a time) to each target
1 table, and finally commits its work. Contrast with “transaction-mode processing” on page 87.

1 **table queue.** A mechanism for transferring rows between database partitions. Table queues are distributed row
1 streams with simplified rules for the insertion and removal of rows. Table queues can also be used to deliver rows
1 between different processes in a single-partition database.

table space. (1) An abstraction of a collection of containers into which database objects are stored. A table space
provides a level of indirection between a database and the tables stored within the database. A table space has space
on media storage devices assigned to it. The data, index, long field, and LOB portions of a table can be stored in the
same table space, or can be individually broken out into separate table spaces. (2) In DB2 Universal Database for
z/OS and OS/390, a page set that is used to store the records in one or more tables.

1 **table space container.** An allocation of space to a table space. Depending on the table space type, the container can
1 be a directory, device, or file.

1 **table space set.** In DB2 Universal Database for z/OS and OS/390, a set of table spaces and partitions that should be
1 recovered together if each contains a table that is a parent or descendent of a table in one of the others, or the set that
1 contains a base table and associated auxiliary tables. A table space set can contain both types of relationships.

7 **takeover.** In high availability disaster recovery (HADR), the process by which the standby database becomes the
7 new primary database.

tag. An element of the tag language. Tags indicate actions to be taken when the tag language file is imported to the
information catalog.

tag language. A format for defining object types and objects, and actions to be taken on those object types and
objects, in the Data Warehouse Center or the information catalog.

tag language file. A file that contains tag language that describes objects and object types to be added, updated or
deleted in the Data Warehouse Center or in the information catalog, when the file is imported.

In the Information Catalog Center, a tag language file is produced when you:

- Transfer a delete history log.
- Extract descriptive data from another database system using an extract program.

1 **target.** In the Data Warehouse Center, a table, view, or file that is produced or populated by a step; the output of a
1 step. See also “source” on page 77.

7 **target server.** (1) In SQL replication, a database or subsystem that contains replication target tables, views, or stored
 7 procedures. (2) In Q replication, a database or subsystem that contains replication target tables or stored procedures.
 7 Compare with “Q Apply server” on page 65.

7 **target table.** (1) In SQL replication, a table that is the destination for changes from a registered replication source. A
 7 target table can be a user copy table, a point-in-time table, a base aggregate table, a change aggregate table, a CCD
 7 table, or a replica table. (2) In Q replication, a table that is the destination for replicated changes from a source that is
 7 part of a Q subscription.

| **task.** In the Task Center, a unit of work and its associated schedule and task actions. Tasks can be set to run on
 | schedules and can perform various actions based on the success or failure of the task. DB2 Universal Database
 | scripts, operating scripts, and warehouse steps are all examples of tasks. See also “task action” and “step” on page
 | 80.

| **task action.** In the Task Center, an action that is performed based on the completion status of a particular task. For
 | example, “If Task A completes successfully, run Task B,” and “If Task Z fails, disable the schedule of Task Y.” See also
 | “task” and “step” on page 80.

| **task category.** A string that is associated with any number of tasks in the Task Center for easier administration of
 | related tasks. For example, you can create a task category named “Payroll” then group all payroll-related tasks in the
 | Payroll category.

| **Task Center.** The DB2 Universal Database graphical interface for organizing task flow, scheduling tasks, and
 | distributing notifications about the status of completed tasks.

task control block (TCB). A control block that is used to communicate information about tasks within an address
 space that are connected to DB2 Universal Database for z/OS and OS/390. An address space can support many task
 connections (as many as one per task), but only one address space connection.

TCB. See “task control block.”

TCP/IP. See “Transmission Control Protocol/Internet Protocol” on page 88.

| **TCP/IP port.** A 2-byte value that identifies a TCP/IP network application within a TCP/IP host.

| **technical metadata.** In the Data Warehouse Center, data that describes the technical aspects of the data, such as its
 | database type and length. Technical metadata includes information about where the data comes from and the rules
 | used to extract, clean, and transform the data. Much of the metadata in the Data Warehouse Center is technical. See
 | also “business metadata” on page 9.

| **template.** In a z/OS or OS/390 environment, a DB2 Universal Database utilities output data set descriptor that is
 | used for dynamic allocation. A template is defined by the TEMPLATE utility control statement.

| **temporary table.** A table that holds temporary data. For example, temporary tables are useful for holding or sorting
 | intermediate results from queries that contain a large number of rows. The two kinds of temporary tables, which are
 | created by different SQL statements, are the created temporary table and the declared temporary table. See also
 | “result table” on page 72, “created temporary table” on page 19, and “declared temporary table” on page 26.

temporary table space. A table space that can store only temporary tables.

7 **territory.** A portion of the POSIX locale that is mapped to the territory code for internal processing by the database
 7 manager.

| **territory code.** A code that is used by DB2 Universal Database to preset the default collation order for an SBCS
 | database and to establish monetary, date, time, and numeric formatting that is specific to a country, region, or
 | territory.

7 **tessellation.** The division of a surface into a mesh or network.

| **thread.** (1) The database manager structure that describes an application’s connection, traces its progress, processes
 | resource functions, and delimits its accessibility to the database manager resources and services. Most DB2 Universal
 | Database for z/OS and OS/390 functions execute under a thread structure. See also “allied thread” on page 3 and
 | “database access thread” on page 20. (2) In some operating systems, the smallest unit of operation to be performed in
 | a process.

Glossary

three-part name. The full name of a table, view, or alias that consists of a location name, authorization identifier, and an object name, separated by periods.

threshold trigger. An event that occurs when the value of a performance variable exceeds or falls below a user-defined threshold value. The action that occurs as a result of a threshold trigger can be:

- Logging information in an alert log file.
- Displaying information in an alert log window.
- Generating an audio alarm.
- Issuing a message window.
- Invoking a predefined command or program.

throttled utilities. Utilities that have a limit placed on the resources that would otherwise be consumed. The degree to which the resources are limited is based on the current workload of the system. Supported utilities include backup, restore, and table space reorganization.

time. A three-part value that designates a time of day in hours, minutes, and seconds.

time duration. A DECIMAL(6,0) value that represents a number of hours, minutes, and seconds.

timeout. An abnormal termination of either the DB2 Universal Database for z/OS and OS/390 subsystem or of an application because of the unavailability of resources. Installation specifications are set to determine both the amount of time DB2 Universal Database for z/OS and OS/390 waits for IRLM services after starting, and the amount of time IRLM waits if a resource that an application requests is unavailable. If either of these time specifications is exceeded, a timeout is declared.

timeron. A unit of measurement that is used to give a rough relative estimate of the resources, or cost, required by the database server to execute two plans for the same query. The resources calculated in the estimate include weighted processor and I/O costs.

Time-Sharing Option (TSO). In a z/OS or OS/390 environment, software that provides interactive communications, allowing a user or programmer to start an application from a terminal and work with the application. TSO is required for binding application plans and packages and for executing several online functions that are provided with DB2 Universal Database for z/OS and OS/390.

timestamp. A data type that contains a seven-part value that consists of a date and time expressed in years, months, days, hours, minutes, seconds, and microseconds.

timestamp duration. A DECIMAL(20,6) value that represents a number of years, months, days, hours, minutes, seconds, and microseconds.

Tivoli Space Manager. A feature of the Tivoli Storage Manager product that moves files in and out of a secondary storage medium based upon actual file accesses in the primary native file system. This feature can be used with DB2 Data Links Manager to enable DATALINK files to be stored in a virtually infinitely sized file system.

Tivoli Storage Manager (TSM). A client/server product that provides storage management and data access services in a heterogeneous environment. TSM supports various communication methods, provides administrative facilities to manage the backup and storage of files, and provides facilities for scheduling backups.

TM Database. See "Transaction Manager Database" on page 87.

to-do. A state of a unit of recovery that indicates that the changes by the unit of recovery to recoverable DB2 Universal Database for z/OS and OS/390 resources are indoubt and must be either applied to the DASD media or backed out, as determined by the commit coordinator.

token. The basic syntactic unit of a computing language. A token consists of one or more characters, excluding the blank character and excluding characters within a string constant or delimited identifier.

topology and routing services (TRS). An APPN control point component that manages the topology database and computes routes.

total time. See "elapsed total time" on page 31.

TP. See "transaction program" on page 87.

7 **trace.** (1) A DB2 Universal Database for z/OS and OS/390 facility that is used to monitor and collect auditing,
7 performance, accounting, statistics, and serviceability (global) data. (2) In DB2 replication, a facility that is used to
7 collect monitoring, auditing, and performance data for the Capture program, the Q Capture program, Apply
7 program, the Q Apply program, or the Replication Alert Monitor.

7 **transaction.** (1) An atomic series of SQL statements that make up a logical unit of work. All of the data
7 modifications made during a transaction are either committed together as a unit or are all rolled back as a unit.
7 Synonym for “unit of work” on page 90. (2) An exchange between a server and a program, two servers, or two
7 programs that accomplishes a particular action or result. An example of a transaction is the entry of a customer’s
7 deposit and the subsequent update of the customer’s balance.

| **transaction-based replication.** In SQL replication, a type of processing in which every transaction is replicated to the
| target table when it is committed in the source table. Contrast with “transaction-consistent replication.”

transaction compensation. A process that restores rows that are affected by a committed transaction that is rejected. When a committed transaction is rejected, the rows are restored to the state that they were in before the transaction was committed.

| **transaction-consistent replication.** In SQL replication, a type of processing in which the net result of all transaction
| updates is replicated to the target table. Contrast with “transaction-based replication.”

transaction lock. In DB2 Universal Database for z/OS and OS/390, a lock that is used to control concurrent execution of SQL statements.

transaction manager. A function that assigns identifiers to transactions, monitors their progress, and takes responsibility for transaction completion and failure recovery.

Transaction Manager Database (TM Database). A database that is used to log transactions when a two-phase commit (SYNCPPOINT TWOPHASE) is used with DB2 databases. In the event of transaction failure, the TM Database information can be accessed to resynchronize databases involved in the failed transaction.

| **transaction-mode processing.** In SQL replication, a type of replication subscription-set processing in which the
| Apply program retrieves data from the source CD table, then applies the data to the target table in the same commit
| sequence that is used at the source. The Apply program processes transactions for all subscription-set members
| together, rather than sequentially. Contrast with “table-mode processing” on page 84.

transaction program (TP). An application program that uses APPC to communicate with a partner application program.

transaction program name. In SNA LU 6.2 conversations, the name of the program at the remote logical unit that is to be the other half of the conversation.

transformation. In the Data Warehouse Center, an operation performed on data. Pivot and cleanse are types of transformations.

| **transformation relationship category.** In the Information Catalog Center, a category for relationship types that
| connects transformation objects to data resources. For example, you can connect a Transformation object to a File
| object. Objects that are connected with this category of relationship are displayed in the Information Catalog Center
| Show Lineage Tree window.

| **transformer.** A program that operates on warehouse data. The Data Warehouse Center provides two types of
| transformers: statistical transformers, which provide statistics about the data in one or more tables; and warehouse
| transformers, which prepare the data for analysis. Transformers have corresponding step types for the types of data
| manipulation that the steps perform; for example, a clean step uses the Clean Data transformer.

| **transition table.** A temporary table that contains all the affected rows of the subject table in their state before or
| after the triggering event occurs. Triggered SQL statements in the trigger definition can reference the table of changed
| rows in the old state or the new state.

transition variable. A variable that is valid only in FOR EACH ROW triggers. It allows access to the transition values for the current row. An old transition variable is the value of the row before the modification is applied, and the new transition variable is the value of the row after the modification is applied.

Glossary

- 7 **Transmission Control Protocol/Internet Protocol (TCP/IP).** An industry-standard, nonproprietary set of
7 communications protocols that provide reliable end-to-end connections between applications over interconnected
7 networks of different types.
- 7 **transparent DDL statement.** A DDL statement that can be issued outside of a pass-through session to create and
7 modify remote tables.
- | **tree view.** A view that provides a hierarchical view of an object and the objects that it contains.
- | **trigger.** A database object that is associated with a single base table or view and that defines a rule. The rule consists
| of a set of SQL statements that run when an insert, update, or delete database operation occurs on the associated
| base table or view. See also “insert trigger” on page 44 and “instead of trigger” on page 44.
- | **trigger activation.** The process that occurs when the trigger event that is defined in a trigger definition is executed.
| Trigger activation consists of the evaluation of the triggered action condition and conditional execution of the
7 triggered SQL statements. See also “trigger,” “trigger event,” and “trigger activation time.”
- 7 **trigger activation time.** In a trigger definition, the specification of when the trigger should be activated with respect
7 to the trigger event. The trigger activation time can be either before or after the trigger event. See also “trigger,”
7 “trigger event,” “trigger activation,” “before trigger” on page 7, and “after trigger” on page 2.
- 7 **trigger body.** See “triggered SQL statements.”
- | **trigger cascading.** The process that occurs when the triggered action of a trigger causes the activation of another
| trigger.
- 7 **triggered action.** The SQL logic that is run when a trigger event occurs and a trigger is activated. The triggered
7 action consists of an optional triggered action condition and a set of triggered SQL statements that are run only if the
7 triggered action condition is true. See also “trigger,” “trigger event,” “triggered action condition,” and “triggered SQL
7 statements.”
- 7 **triggered action condition.** An optional Boolean search condition in a triggered action, implemented as a WHEN
7 clause, that DB2 Universal Database evaluates to determine whether the triggered SQL statements of the triggered
7 action should be run. See also “trigger,” “triggered action,” and “triggered SQL statements.”
- 7 **triggered SQL statements.** The set of SQL statements that are run when a trigger is activated and its triggered
7 action condition is true. Triggered SQL statements are also referred to as the *trigger body*. See also “trigger,”
7 “triggered action,” and “triggered action condition.”
- 7 **trigger event.** In a CREATE TRIGGER statement, the specification of an insert, update, or delete operation on a
7 specified table that activates the trigger. See also “trigger,” “trigger activation,” and “trigger activation time.”
- 7 **trigger granularity.** In a trigger definition, the specification of how often the trigger is to be activated for an instance
7 of the trigger event. The trigger granularity can be specified to be either once for the triggering SQL statement or
7 once for each row that the triggering SQL statement modifies. See also “trigger” and “trigger event.”
- | **trigger package.** A package that is created when a CREATE TRIGGER statement is executed. The package is
| executed when the trigger is activated.
- | **triggering SQL operation.** The SQL operation that causes a trigger to be activated when performed on the subject
| table.
- truncation.** The process of discarding part of a result from an operation when it exceeds memory or storage
capacity.
- 7 **TSM.** See “Tivoli Storage Manager” on page 86.
- | **TSO.** See “Time-Sharing Option” on page 86.
- TSO attachment facility.** A DB2 Universal Database for z/OS and OS/390 facility that consists of the DSN
command processor and DB2I. Applications that are not written for the CICS or IMS environments can run under the
TSO attachment facility.

- | **tuning parameters table.** A table at the source server that contains timing information used by the Capture program. The information includes how long to keep rows in the change data table; how much time can elapse before changes are stored in a database log or journal; how often to commit changed data to the unit-of-work tables. See also “tuple.”
- | **tuple.** A synonym for a row in a table. See also “tuning parameters table.”
- | **two-phase commit.** A two-step process by which recoverable resources and an external subsystem are committed. During the first step, the database manager subsystems are polled to ensure that they are ready to commit. If all subsystems respond positively, the database manager instructs them to commit. See also “distributed transaction” on page 29.
- | **typed parameter marker.** A parameter marker that is specified along with its target data type. It has the general form: *CAST (? AS data-type)*.
- | **typed table.** A table in which the data type of each column is defined separately or the types for the columns are based on the attributes of a user-defined structured type.
- | **typed view.** A view in which the data type of each column is derived from the result table, or the types for the columns are based on the attributes of a user-defined structure type.
- | **type 1 index.** An index that is not a type 2 index. As of DB2 Universal Database for z/OS and OS/390 Version 8, type 1 indexes are no longer supported. See also “type 2 indexes”
- | **type 2 index.** A pseudo delete index that supports variable length key parts with length greater than 255. See also “type 1 indexes.”

U

- | **UCS-2.** Universal Character Set, coded in 2 octets, which means that characters are represented in 16-bits per character.
- 7 **UDDI.** See “Universal Description, Discovery, and Integration” on page 90.
- | **UDF.** See “user-defined function” on page 91.
- | **UDT.** See “user-defined type” on page 91.
- | **UFS.** See “UNIX File System (UFS)” on page 90.
- | **unambiguous cursor.** A cursor that allows a DBMS to determine whether blocking can be used with the answer set. A cursor that is defined FOR FETCH ONLY or FOR READ ONLY can be used with blocking, whereas a cursor that is defined FOR UPDATE cannot. See also “ambiguous cursor” on page 3.
- | **unbind session (UNBIND).** A request to deactivate a session between two logical units (LUs).
- | **uncommitted read (UR).** An isolation level that allows an application to access uncommitted changes of other transactions. The application does not lock other applications out of the row that it is reading, unless the other application attempts to drop or alter the table. See also “repeatable read” on page 70, “cursor stability” on page 19, and “read stability” on page 67.
- | **underlying view.** In DB2 Universal Database for z/OS and OS/390, the view on which another view is directly or indirectly defined.
- | **undo.** (1) To recover the last edit that has taken place. (2) A state of a unit of recovery that indicates that the changes that the unit of recovery made to recoverable DB2 Universal Database for z/OS and OS/390 resources must be backed out.
- | **Unicode.** An international character encoding scheme that is a subset of the ISO 10646 standard. Each character supported is defined using a unique 2-byte code. See also “ASCII” on page 5 and “EBCDIC” on page 31.
- 7 **unidirectional replication.** In Q replication, a configuration in which changes that occur at a source table are replicated over WebSphere MQ queues to a target table or are passed to a stored procedure to manipulate the data.
- 7 Changes that occur at the target table are not replicated back to the source table.

Glossary

| **uniform resource locator (URL).** A sequence of characters that represents information resources on a computer or in
| a network such as the Internet. This sequence of characters includes the abbreviated name of the protocol that is used
| to access the information resource and the information that is used by the protocol to locate the information resource.

| **union.** An SQL operation that combines the results of two select statements. Unions are often used to merge lists of
| values that are obtained from several tables.

| **unique constraint.** The rule that no two values in a primary key or key of a unique index can be the same. Also
7 referred to as *uniqueness constraint*. See also “constraint” on page 16, “check constraint” on page 11, “referential
7 constraint” on page 68, and “informational constraint” on page 43.

| **unique identifier (UI).** In the Information Catalog Center, a key for an object. The key is comprised of up to 16
| properties, which, when concatenated in a designated order, uniquely identify the object during the import function.

| **unique index.** An index that ensures that no identical key values are stored in a table.

| **unique key.** A key that is constrained so that no two of its values are equal.

| **unit of recovery.** A recoverable sequence of operations within a single resource manager, such as an instance of DB2
Universal Database for z/OS and OS/390. See also “unit of work.”

| **unit of work (UOW).** A recoverable sequence of operations within an application process. At any time, an
application process is a single unit of work, but the life of an application process can involve many units of work as
a result of commit or rollback operations. In a DB2 Universal Database for z/OS and OS/390 *multisite update*
operation, a single unit of work can include several *units of recovery*. Synonym for “transaction” on page 87. See also
7 “unit of recovery” and “multisite update” on page 54.

7 **unit-of-work table (UOW table).** In SQL replication, a control table that is stored in the Capture control server that
7 contains commit records that are read from the database log or journal. The records show that a transaction or unit of
7 work committed successfully and include a unit-of-recovery ID that can be used to join the unit-of-work table and
7 the CD table to produce transaction-consistent change data.

7 **Universal Description, Discovery, and Integration (UDDI).** A mechanism for clients to dynamically find other Web
7 services.

7 **Universal Time Coordinated.** The international time standard. 00:00 UTC is midnight in Greenwich, England.

| **UNIX File System (UFS).** The native file system in the Solaris Operating Environment.

| **unlink.** The action that DB2 Data Links Manager takes to give up control of a file that is no longer referenced in a
| table that contains a DATALINK column. A file can be unlinked as the result of such database actions as an SQL
| UPDATE, DELETE, or DROP TABLE.

| **unlinked file.** In a DB2 Data Links Manager environment, a file that is controlled by the native file system on an
| operating system. By contrast, a linked file is controlled by the DLFF component.

| **unlock.** To release an object or system resource that was previously locked and return it to general availability
within DB2 Universal Database for z/OS and OS/390.

| **untyped parameter marker.** A parameter marker that is specified without its target data type. It has the form of a
single question mark.

| **updatability.** The ability of a cursor to perform positioned updates and deletes. The updatability of a cursor can be
| influenced by the SELECT statement and the cursor sensitivity option that is specified on the DECLARE CURSOR
| statement.

7 **updatable result set.** A result set that is associated with a cursor that was created with a SELECT statement that
7 contains the FOR UPDATE clause. See also “result set” on page 72.

| **update.** In a DB2 Data Links Manager environment, to modify a linked file.

| **update-anywhere replication.** In SQL replication, a configuration in which all tables are both registered sources and
| read-write targets. One table is the primary source table for full refresh of all the others. Contrast with “peer-to-peer
| replication” on page 61. See also “multi-tier replication” on page 54, “master table” on page 52, and “replica table”
| on page 70.

| **update hole.** A row for a SELECT statement of a cursor that no longer has a corresponding row in the base table because the row was updated. An update hole is created when a row in the base table is updated such that the row no longer qualifies to be in the result set while a cursor is open whose SELECT statement result contains the row that is updated. Such a row is no longer accessible through the cursor. See also “delete hole” on page 26.

| **update-in-place.** In a DB2 Data Links Manager environment, the process of making changes to a linked file while a DATALINK column value in a database is pointing to that file. Any changes to linked files during an update-in-place operation become visible to database users when the DB2 Universal Database host is notified that the update is complete.

| **update-in-progress state.** The logical state of a file under the control of a DB2 Data Links Manager that is in the process of being updated. A linked file enters this state after it is opened using a write token, and is no longer in this state when the DB2 Universal Database host is notified that the update is complete.

update rule. A condition enforced by the database manager that must be met before a column can be updated.

7 **update trigger.** In DB2 Universal Database for z/OS and OS/390, a trigger that is activated when an update operation occurs on the base table of the trigger definition. See also “trigger” on page 88 and “trigger activation” on page 88.

upstream. In DB2 Universal Database for z/OS and OS/390, the node in the syncpoint tree that is responsible, in addition to other recovery or resource managers, for initiating the execution of a two-phase commit.

UR. See “uncommitted read (UR)” on page 89.

URID (unit of recovery ID). In DB2 Universal Database for z/OS and OS/390, the LOGRBA of the first log record for a unit of recovery. The URID also appears in all subsequent log records for that unit of recovery.

| **URL.** See “uniform resource locator” on page 90.

| **user.** In the Information Catalog Center, a person who accesses the information available in the information catalog but who is not an administrator. Some users can also perform object management tasks normally performed by administrators, such as creating and updating objects. See also “administrator” on page 2 and “power user” on page 62.

7 **user copy table.** In SQL replication, a replication target table whose content matches all or part of a registered source table and contains only user data columns.

user-defined data type. See “distinct type” on page 28.

user-defined distinct type. See “distinct type” on page 28.

7 **user-defined function (UDF).** A database object that is created with the CREATE FUNCTION statement. All functions that are not built-in functions are user-defined functions. See also “function” on page 36 and “built-in function” on page 9.

user-defined performance variable. A performance variable that is created by a user and added to the performance variable profile.

user-defined program. A program that a user supplies and defines to the Data Warehouse Center, as contrasted with supplied programs, which are included with and defined automatically in the Data Warehouse Center.

| **user-defined structured type.** See “structured type” on page 81.

| **user-defined type (UDT).** A data type that is not native to the database manager and was created by a user. In DB2 Universal Database, the term distinct type is used instead of user-defined type.

| **user exit program.** A program, written by a user, that receives control at predefined user exit points. When a user exit program is invoked, the database manager passes control to the executable file. Only one user exit program can be invoked in a database manager instance.

| **user mapping.** In a federated system, the association between the authorization ID at the federated server and the authorization ID at the data source. User mappings are needed so that distributed requests can be sent to the data source. User mappings are created when a user’s authorization ID to access the federated database differs from the

Glossary

| user's authorization ID to access a data source. The CREATE USER MAPPING statement is used to define the association. The ALTER USER MAPPING statement is used to modify a user mapping that you have already created.

| **user options.** In a federated system, parameters of the CREATE USER MAPPING and ALTER USER MAPPING statements to which values related to authorization are assigned. For example, suppose that a user has the same ID with different passwords for the federated database and a data source. For the user to access the data source, it is necessary to map the passwords to one another. This is accomplished with the user option REMOTE_PASSWORD. See "user mapping" on page 91.

7 **user table.** In SQL replication, a table created for and used by an application before it is defined as a replication source. A user table is used as the source for updates to read-only target tables, consistent-change-data tables, replicas, and row-replica tables.

2 **user time.** In UNIX, represents the time spent executing database manager code. See also "system time" on page 83.

| **user view.** In logical data modeling, a model or representation of critical information that the business requires.

UTC. See "Universal Time Coordinated" on page 90.

| **UTF-8.** Unicode Transformation Format, 8-bit encoding form, which is designed for ease of use with existing ASCII-based systems. The CCSID value for data in UTF-8 format is 1208. DB2 Universal Database for z/OS and OS/390 supports UTF-8 in mixed data fields.

| **UTF-16.** Unicode Transformation Format, 16-bit encoding form, which is designed to provide code values for over a million characters and is a superset of UCS-2. The CCSID value for data in UTF-16 format is 1200. DB2 Universal Database for z/OS and OS/390 supports UTF-16 in graphic data fields.

V

| **value.** (1) The alpha or numeric content of a field or variable. (2) The smallest unit of data manipulated in SQL. (3) A specific data item at the intersection of a column and a row.

variable. A data element that specifies a value that can be changed. See also "constant" on page 16.

7 **variant function.** A user-defined function whose result is dependent on its input parameter values as well as other factors. Successive invocations with the same parameter values might produce different results. See also "not deterministic function" on page 56.

| **variable-length string.** A character, graphic, or binary string whose length is not fixed but can range within set limits. Also referred to as a *varying length string*.

| **vectored I/O.** See "scattered read" on page 74.

| **version.** In DB2 Universal Database for z/OS and OS/390, a member of a set of similar programs, DBRMs, packages, or LOBs. Some examples are:

- | • A version of a program is the source code that is produced by precompiling the program. The program version is identified by the program name and a timestamp (consistency token).
- | • A version of a DBRM is the DBRM that is produced by precompiling a program. The DBRM version is identified by the same program name and timestamp as a corresponding program version.
- | • A version of a package is the result of binding a DBRM within a particular database system. The package version is identified by the same program name and consistency token as the DBRM.
- | • A version of a LOB is a copy of a LOB value at a point in time. The version number for a LOB is stored in the auxiliary index entry for the LOB.

| **version recovery.** The restoration of a previous version of the database, using an image that was created during a backup operation. See also "crash recovery" on page 18 and "rollforward recovery" on page 72.

| **view.** (1) A logical table that consists of data that is generated by a query. A view is based on an underlying set of base tables, and the data in a view is determined by a SELECT statement that is run on the base tables. Contrast with base table. (2) A way of looking at the information about, or contained in objects. Each view might reveal different information about its objects. See also "base table" on page 7.

view check option. In DB2 Universal Database for z/OS and OS/390, an option that specifies whether every row that is inserted or updated through a view must conform to the definition of that view. A view check option can be specified with the WITH CASCADED CHECK OPTION, WITH CHECK OPTION, or WITH LOCAL CHECK OPTION clauses of the CREATE VIEW statement.

Virtual Storage Access Method (VSAM). An access method for direct or sequential processing of fixed-length and varying-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunications Access Method (VTAM). In an OS/390 environment, an IBM licensed program that controls communication and the flow of data in an SNA network.

Visual Explain. A tool that provides a graphical interface for database administrators and application programmers to display and analyze detailed information on the access plan of a given SQL statement. The tasks provided by this tool can be accessed from the Control Center.

7 **volatile table.** A table for which SQL operations choose index access whenever possible.

7 **Voronoi cell.** A region on the surface of the Earth that has boundaries with neighboring regions. The boundaries are defined by the geodesic distances between the center point of the Voronoi cell and the center points of its neighbors. A Voronoi cell consists of all points that are closer to the center of the Voronoi cell than to the center of any other Voronoi cell.

7 **Voronoi cell structure.** A subdivision of the Earth's surface into cells where every point within a particular cell is closer to the center point of that cell than the center point of any other cell.

VSAM. See "Virtual Storage Access Method."

VTAM. See "Virtual Telecommunications Access Method."

W

warehouse. See "data warehouse" on page 23.

| **warehouse agent.** In the Data Warehouse Center, a run-time process, capable of running on various operating systems, that performs data extraction, transformation, movement, and loading (ETML) and that can also start user programs. See also "warehouse server."

warehouse control database. The Data Warehouse Center database that contains the control tables that are required to store Data Warehouse Center metadata.

warehouse program group. In the Data Warehouse Center, a container (folder) that holds program objects.

| **warehouse server.** In the Data Warehouse Center, the Windows or AIX component that manages and schedules the data extraction, transformation, movement and loading (ETML) tasks run by the warehouse agents. See also "warehouse agent."

warehouse source. A subset of tables and views from a single database, or a set of files, that have been defined to the Data Warehouse Center.

warehouse target. A subset of tables, indexes, and aliases from a single database that are managed by the Data Warehouse Center.

7 **warm start.** In replication, the process of starting the Capture program so that it reads transactions from the point where it left off. Contrast with "cold start" on page 13.

7 **Web service.** A modular application that performs specific tasks and is accessible through open protocols such as HTTP and SOAP.

7 **Web services description language (WSDL).** A set of definitions that consist of service, port, message, bindings, and port type. WSDL provides a way for service providers to describe the basic format of Web service requests over different protocols or encodings.

Glossary

- 7 **WebSphere MQ.** A family of IBM licensed programs that provides message queuing services.
- | **well-known address.** An address that is used to uniquely identify a particular node in the network to establish connections between nodes. The well-known address is a combination of the network address and the port used on the logical node.
- WLM application environment.** An MVS Workload Manager attribute that is associated with one or more stored procedures. The WLM application environment determines the address space in which a given DB2 Universal Database for z/OS and OS/390 stored procedure runs.
- 7 **work file.** In SQL replication, a temporary file that is used by the Apply program when it processes a subscription set.
- | **wrapper.** In a federated system, the mechanism that the federated server uses to communicate with and retrieve data from the data sources. To implement a wrapper, the federated server uses routines stored in a library called a wrapper module. These routines allow the federated server to perform operations such as connecting to a data source and retrieving data from it iteratively. The DB2 Universal Database federated instance owner uses the CREATE WRAPPER statement to register a wrapper for each data source that is to be included in the federated system.
- | **write token.** The authorization key that is required for updating a file that is referenced in a WRITE PERMISSION ADMIN DATALINK column.
- write to operator (WTO).** An optional user-coded service that allows a message to be written to the system console operator, informing the operator of errors and unusual system conditions that might need to be corrected.
- 7 **WSDL.** See “Web services description language” on page 93.
- WTO.** See “write to operator.”
- WTOR.** A write to operator (WTO) with reply.

X

- | **XBSA.** An industry-standard API set for backup and restore utilities. XBSA is one of the archive options that is available for use in maintaining backup copies of linked files in the DB2 Data Links Manager environment. The XBSA option is specified with the DLFM_BACKUP_TARGET registry variable.
- XCF.** See “cross-system coupling facility” on page 19.
- XES.** See “cross-system extended services” on page 19.
- XID.** Exchange station ID.
- | **XML.** See “extensible markup language” on page 33.
- 7 **XML attribute.** A name-value pair within a tagged XML element that modifies certain features of the element.
- | **XML collection.** A collection of relational tables from which XML documents are decomposed or that form content of XML documents to be decomposed.
- | **XML column.** A column that has a type of XML Extender user-defined type and the contents of the column are whole XML documents.
- | **XML element.** Logical structures in XML documents delimited by a start and an end tag. An element can be specified in the DTD by an element type declaration.
- 7 **XML publication.** In event publishing, an object that identifies what changes are published from a source table to a user application. The Q Capture program publishes changes from a source table and puts those changes on a send queue in XML format.
- 7 **XML publishing function.** A function that returns XML values from SQL values.
- | **XML shredder.** A function that parses an XML document, extracting rows of data from an XML table.

XRF. See “extended recovery facility” on page 33.

Z

- | z/OS. An IBM operating system for the IBM eServer product line that supports 64-bit real storage.

DB2 Universal Database technical information

DB2 documentation and help

DB2® technical information is available through the following tools and methods:

- DB2 Information Center
 - Topics
 - Help for DB2 tools
 - Sample programs
 - Tutorials
- Downloadable PDF files, PDF files on CD, and printed books
 - Guides
 - Reference manuals
- Command line help
 - Command help
 - Message help
 - SQL state help
- Installed source code
 - Sample programs

You can access additional DB2 Universal Database™ technical information such as technotes, white papers, and Redbooks™ online at ibm.com®. Access the DB2 Information Management software library site at www.ibm.com/software/data/pubs/.

DB2 documentation updates

7 IBM® may periodically make documentation FixPaks and other documentation
7 updates to the DB2 Information Center available. If you access the DB2
7 Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>, you
7 will always be viewing the most up-to-date information. If you have installed the
7 DB2 Information Center locally, then you need to install any updates manually
7 before you can view them. Documentation updates allow you to update the
7 information that you installed from the *DB2 Information Center CD* when new
7 information becomes available.

7 The Information Center is updated more frequently than either the PDF or the
7 hardcopy books. To get the most current DB2 technical information, install the
7 documentation updates as they become available or go to the DB2 Information
7 Center at the www.ibm.com site.

Related concepts:

- “CLI sample programs” in the *CLI Guide and Reference, Volume 1*
- “Java sample programs” in the *Application Development Guide: Building and Running Applications*
- “DB2 Information Center” on page 98

Related tasks:

- “Invoking contextual help from a DB2 tool” on page 115

- “Updating the DB2 Information Center installed on your computer or intranet server” on page 107
- “Invoking message help from the command line processor” on page 116
- “Invoking command help from the command line processor” on page 117
- “Invoking SQL state help from the command line processor” on page 117

Related reference:

- “DB2 PDF and printed documentation” on page 109

DB2 Information Center

The DB2[®] Information Center gives you access to all of the information you need to take full advantage of DB2 family products, including DB2 Universal Database[™], DB2 Connect[™], DB2 Information Integrator and DB2 Query Patroller[™]. The DB2 Information Center also contains information for major DB2 features and components including replication, data warehousing, and the DB2 extenders.

The DB2 Information Center has the following features if you view it in Mozilla 1.0 or later or Microsoft[®] Internet Explorer 5.5 or later. Some features require you to enable support for JavaScript[™]:

Flexible installation options

You can choose to view the DB2 documentation using the option that best meets your needs:

- To effortlessly ensure that your documentation is always up to date, you can access all of your documentation directly from the DB2 Information Center hosted on the IBM[®] Web site at <http://publib.boulder.ibm.com/infocenter/db2help/>
- To minimize your update efforts and keep your network traffic within your intranet, you can install the DB2 documentation on a single server on your intranet
- To maximize your flexibility and reduce your dependence on network connections, you can install the DB2 documentation on your own computer

Search

7 You can search all of the topics in the DB2 Information Center by entering
 7 a search term in the **Search** text field. You can retrieve exact matches by
 7 enclosing terms in quotation marks, and you can refine your search with
 7 wildcard operators (*, ?) and Boolean operators (AND, NOT, OR).

Task-oriented table of contents

7 You can locate topics in the DB2 documentation from a single table of
 7 contents. The table of contents is organized primarily by the kind of tasks
 7 you may want to perform, but also includes entries for product overviews,
 7 goals, reference information, an index, and a glossary.

- Product overviews describe the relationship between the available products in the DB2 family, the features offered by each of those products, and up to date release information for each of these products.
- Goal categories such as installing, administering, and developing include topics that enable you to quickly complete tasks and develop a deeper understanding of the background information for completing those tasks.

- 7 • Reference topics provide detailed information about a subject, including
7 statement and command syntax, message help, and configuration
7 parameters.

7 **Show current topic in table of contents**

7 You can show where the current topic fits into the table of contents by
7 clicking the **Refresh / Show Current Topic** button in the table of contents
7 frame or by clicking the **Show in Table of Contents** button in the content
7 frame. This feature is helpful if you have followed several links to related
7 topics in several files or arrived at a topic from search results.

Index You can access all of the documentation from the index. The index is
organized in alphabetical order by index term.

Glossary

You can use the glossary to look up definitions of terms used in the DB2
documentation. The glossary is organized in alphabetical order by glossary
term.

7 **Integrated localized information**

7 The DB2 Information Center displays information in the preferred
7 language set in your browser preferences. If a topic is not available in your
7 preferred language, the DB2 Information Center displays the English
7 version of that topic.

For iSeries™ technical information, refer to the IBM eServer™ iSeries information
center at www.ibm.com/eserver/series/infocenter/.

Related concepts:

- “DB2 Information Center installation scenarios” on page 99

Related tasks:

- “Updating the DB2 Information Center installed on your computer or intranet server” on page 107
- “Displaying topics in your preferred language in the DB2 Information Center” on page 108
- “Invoking the DB2 Information Center” on page 106
- “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” on page 102
- “Installing the DB2 Information Center using the DB2 Setup wizard (Windows)” on page 104

7 **DB2 Information Center installation scenarios**

7 Different working environments can pose different requirements for how to access
7 DB2® information. The DB2 Information Center can be accessed on the IBM® Web
7 site, on a server on your organization’s network, or on a version installed on your
7 computer. In all three cases, the documentation is contained in the DB2
7 Information Center, which is an architected web of topic-based information that
7 you view with a browser. By default, DB2 products access the DB2 Information
7 Center on the IBM Web site. However, if you want to access the DB2 Information
7 Center on an intranet server or on your own computer, you must install the DB2
7 Information Center using the DB2 Information Center CD found in your product
7 Media Pack. Refer to the summary of options for accessing DB2 documentation
7 which follows, along with the three installation scenarios, to help determine which

method of accessing the DB2 Information Center works best for you and your work environment, and what installation issues you might need to consider.

Summary of options for accessing DB2 documentation:

The following table provides recommendations on which options are possible in your work environment for accessing the DB2 product documentation in the DB2 Information Center.

Internet access	Intranet access	Recommendation
Yes	Yes	Access the DB2 Information Center on the IBM Web site, or access the DB2 Information Center installed on an intranet server.
Yes	No	Access the DB2 Information Center on the IBM Web site.
No	Yes	Access the DB2 Information Center installed on an intranet server.
No	No	Access the DB2 Information Center on a local computer.

Scenario: Accessing the DB2 Information Center on your computer:

Tsu-Chen owns a factory in a small town that does not have a local ISP to provide him with Internet access. He purchased DB2 Universal Database™ to manage his inventory, his product orders, his banking account information, and his business expenses. Never having used a DB2 product before, Tsu-Chen needs to learn how to do so from the DB2 product documentation.

After installing DB2 Universal Database on his computer using the typical installation option, Tsu-Chen tries to access the DB2 documentation. However, his browser gives him an error message that the page he tried to open cannot be found. Tsu-Chen checks the installation manual for his DB2 product and discovers that he has to install the DB2 Information Center if he wants to access DB2 documentation on his computer. He finds the *DB2 Information Center CD* in the media pack and installs it.

From the application launcher for his operating system, Tsu-Chen now has access to the DB2 Information Center and can learn how to use his DB2 product to increase the success of his business.

Scenario: Accessing the DB2 Information Center on the IBM Web site:

Colin is an information technology consultant with a training firm. He specializes in database technology and SQL and gives seminars on these subjects to businesses all over North America using DB2 Universal Database. Part of Colin's seminars includes using DB2 documentation as a teaching tool. For example, while teaching courses on SQL, Colin uses the DB2 documentation on SQL as a way to teach basic and advanced syntax for database queries.

Most of the businesses at which Colin teaches have Internet access. This situation influenced Colin's decision to configure his mobile computer to access the DB2 Information Center on the IBM Web site when he installed the latest version of DB2 Universal Database. This configuration allows Colin to have online access to the latest DB2 documentation during his seminars.

7 However, sometimes while travelling Colin does not have Internet access. This
7 posed a problem for him, especially when he needed to access to DB2
7 documentation to prepare for seminars. To avoid situations like this, Colin installed
7 a copy of the DB2 Information Center on his mobile computer.

7 Colin enjoys the flexibility of always having a copy of DB2 documentation at his
7 disposal. Using the **db2set** command, he can easily configure the registry variables
7 on his mobile computer to access the DB2 Information Center on either the IBM
7 Web site, or his mobile computer, depending on his situation.

7 **Scenario: Accessing the DB2 Information Center on an intranet server:**

7 Eva works as a senior database administrator for a life insurance company. Her
7 administration responsibilities include installing and configuring the latest version
7 of DB2 Universal Database on the company's UNIX[®] database servers. Her
7 company recently informed its employees that, for security reasons, it would not
7 provide them with Internet access at work. Because her company has a networked
7 environment, Eva decides to install a copy of the DB2 Information Center on an
7 intranet server so that all employees in the company who use the company's data
7 warehouse on a regular basis (sales representatives, sales managers, and business
7 analysts) have access to DB2 documentation.

7 Eva instructs her database team to install the latest version of DB2 Universal
7 Database on all of the employee's computers using a response file, to ensure that
7 each computer is configured to access the DB2 Information Center using the host
7 name and the port number of the intranet server.

7 However, through a misunderstanding Migual, a junior database administrator on
7 Eva's team, installs a copy of the DB2 Information Center on several of the
7 employee computers, rather than configuring DB2 Universal Database to access the
7 DB2 Information Center on the intranet server. To correct this situation Eva tells
7 Migual to use the **db2set** command to change the DB2 Information Center registry
7 variables (DB2_DOCHOST for the host name, and DB2_DOCPORT for the port
7 number) on each of these computers. Now all of the appropriate computers on the
7 network have access to the DB2 Information Center, and employees can find
7 answers to their DB2 questions in the DB2 documentation.

7 **Related concepts:**

- 7 • "DB2 Information Center" on page 98

7 **Related tasks:**

- 7 • "Updating the DB2 Information Center installed on your computer or intranet
7 server" on page 107
- 7 • "Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)" on
7 page 102
- 7 • "Installing the DB2 Information Center using the DB2 Setup wizard (Windows)"
7 on page 104
- 7 • "Setting the location for accessing the DB2 Information Center: Common GUI
7 help"

7 **Related reference:**

- 7 • "db2set - DB2 Profile Registry Command" in the *Command Reference*

7 Installing the DB2 Information Center using the DB2 Setup wizard 7 (UNIX)

7 DB2 product documentation can be accessed in three ways: on the IBM Web site,
7 on an intranet server, or on a version installed on your computer. By default, DB2
7 products access DB2 documentation on the IBM Web site. If you want to access the
7 DB2 documentation on an intranet server or on your own computer, you must
7 install the documentation from the *DB2 Information Center CD*. Using the DB2
7 Setup wizard, you can define your installation preferences and install the DB2
7 Information Center on a computer that uses a UNIX operating system.

7 Prerequisites:

7 This section lists the hardware, operating system, software, and communication
7 requirements for installing the DB2 Information Center on UNIX computers.

- 7 • **Hardware requirements**

7 You require one of the following processors:

- 7 – PowerPC (AIX)
- 7 – HP 9000 (HP-UX)
- 7 – Intel 32-bit (Linux)
- 7 – Solaris UltraSPARC computers (Solaris Operating Environment)

- 7 • **Operating system requirements**

7 You require one of the following operating systems:

- 7 – IBM AIX 5.1 (on PowerPC)
- 7 – HP-UX 11i (on HP 9000)
- 7 – Red Hat Linux 8.0 (on Intel 32-bit)
- 7 – SuSE Linux 8.1 (on Intel 32-bit)
- 7 – Sun Solaris Version 8 (on Solaris Operating Environment UltraSPARC
7 computers)

7 **Note:** The DB2 Information Center runs on a subset of the UNIX operating
7 systems on which DB2 clients are supported. It is therefore recommended
7 that you either access the DB2 Information Center from the IBM Web site,
7 or that you install and access the DB2 Information Center on an intranet
7 server.

- 7 • **Software requirements**

7 – The following browser is supported:

- 7 – Mozilla Version 1.0 or greater

- 7 • The DB2 Setup wizard is a graphical installer. You must have an implementation
7 of the X Window System software capable of rendering a graphical user
7 interface for the DB2 Setup wizard to run on your computer. Before you can run
7 the DB2 Setup wizard you must ensure that you have properly exported your
7 display. For example, enter the following command at the command prompt:

```
7 export DISPLAY=9.26.163.144:0.
```

- 7 • **Communication requirements**

- 7 – TCP/IP

7 Procedure:

7 To install the DB2 Information Center using the DB2 Setup wizard:

- 7 1. Log on to the system.
- 7 2. Insert and mount the DB2 Information Center product CD on your system.
- 7 3. Change to the directory where the CD is mounted by entering the following
7 command:
7

```
cd /cd
```
- 7 where */cd* represents the mount point of the CD.
- 7 4. Enter the **./db2setup** command to start the DB2 Setup wizard.
- 7 5. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation
7 of the DB2 Information Center, click **Install Product**. Online help is available
7 to guide you through the remaining steps. To invoke the online help, click
7 **Help**. You can click **Cancel** at any time to end the installation.
- 7 6. On the **Select the product you would like to install** page, click **Next**.
- 7 7. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup
7 wizard will guide you through the program setup process.
- 7 8. To proceed with the installation, you must accept the license agreement. On
7 the **License Agreement** page, select **I accept the terms in the license**
7 **agreement** and click **Next**.
- 7 9. Select **Install DB2 Information Center on this computer** on the **Select the**
7 **installation action** page. If you want to use a response file to install the DB2
7 Information Center on this or other computers at a later time, select **Save your**
7 **settings in a response file**. Click **Next**.
- 7 10. Select the languages in which the DB2 Information Center will be installed on
7 **Select the languages to install** page. Click **Next**.
- 7 11. Configure the DB2 Information Center for incoming communication on the
7 **Specify the DB2 Information Center port** page. Click **Next** to continue the
7 installation.
- 7 12. Review the installation choices you have made in the **Start copying files** page.
7 To change any settings, click **Back**. Click **Install** to copy the DB2 Information
7 Center files onto your computer.

7 You can also install the DB2 Information Center using a response file.

7 The installation logs `db2setup.his`, `db2setup.log`, and `db2setup.err` are located, by
7 default, in the `/tmp` directory.

7 The `db2setup.log` file captures all DB2 product installation information, including
7 errors. The `db2setup.his` file records all DB2 product installations on your
7 computer. DB2 appends the `db2setup.log` file to the `db2setup.his` file. The
7 `db2setup.err` file captures any error output that is returned by Java, for example,
7 exceptions and trap information.

7 When the installation is complete, the DB2 Information Center will be installed in
7 one of the following directories, depending upon your UNIX operating system:

- 7 • AIX: `/usr/opt/db2_08_01`
- 7 • HP-UX: `/opt/IBM/db2/V8.1`
- 7 • Linux: `/opt/IBM/db2/V8.1`
- 7 • Solaris Operating Environment: `/opt/IBM/db2/V8.1`

7 **Related concepts:**

- 7 • “DB2 Information Center” on page 98
- 7 • “DB2 Information Center installation scenarios” on page 99

7
7
7
7
7
7
7
7
7
7
7

Related tasks:

- “Installing DB2 using a response file (UNIX)” in the *Installation and Configuration Supplement*
- “Updating the DB2 Information Center installed on your computer or intranet server” on page 107
- “Displaying topics in your preferred language in the DB2 Information Center” on page 108
- “Invoking the DB2 Information Center” on page 106
- “Installing the DB2 Information Center using the DB2 Setup wizard (Windows)” on page 104

7 **Installing the DB2 Information Center using the DB2 Setup wizard**
7 **(Windows)**

7
7
7
7
7
7
7
7
7

DB2 product documentation can be accessed in three ways: on the IBM Web site, on an intranet server, or on a version installed on your computer. By default, DB2 products access DB2 documentation on the IBM Web site. If you want to access the DB2 documentation on an intranet server or on your own computer, you must install the DB2 documentation from the *DB2 Information Center CD*. Using the DB2 Setup wizard, you can define your installation preferences and install the DB2 Information Center on a computer that uses a Windows operating system.

7
7

Prerequisites:

7
7
7

This section lists the hardware, operating system, software, and communication requirements for installing the DB2 Information Center on Windows.

7
7

- **Hardware requirements**

7
7

You require one of the following processors:

- 32-bit computers: a Pentium or Pentium compatible CPU

7
7

- **Operating system requirements**

7
7

You require one of the following operating systems:

- Windows 2000
- Windows XP

7
7
7
7
7

Note: The DB2 Information Center runs on a subset of the Windows operating systems on which DB2 clients are supported. It is therefore recommended that you either access the DB2 Information Center on the IBM Web site, or that you install and access the DB2 Information Center on an intranet server.

7
7

- **Software requirements**

7
7

– The following browsers are supported:

- Mozilla 1.0 or greater
- Internet Explorer Version 5.5 or 6.0 (Version 6.0 for Windows XP)

7
7

- **Communication requirements**

- TCP/IP

7
7

Restrictions:

- You require an account with administrative privileges to install the DB2 Information Center.
- 7

7 **Procedure:**

7 To install the DB2 Information Center using the DB2 Setup wizard:

- 7 1. Log on to the system with the account that you have defined for the DB2
7 Information Center installation.
- 7 2. Insert the CD into the drive. If enabled, the auto-run feature starts the IBM
7 DB2 Setup Launchpad.
- 7 3. The DB2 Setup wizard determines the system language and launches the
7 setup program for that language. If you want to run the setup program in a
7 language other than English, or the setup program fails to auto-start, you can
7 start the DB2 Setup wizard manually.

7 To start the DB2 Setup wizard manually:

- 7 a. Click **Start** and select **Run**.
- 7 b. In the **Open** field, type the following command:
7 `x:\setup.exe /i 2-letter language identifier`

7 where *x*: represents your CD drive, and *2-letter language identifier* represents
7 the language in which the setup program will be run.

- 7 c. Click **OK**.
- 7 4. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation
7 of the DB2 Information Center, click **Install Product**. Online help is available
7 to guide you through the remaining steps. To invoke the online help, click
7 **Help**. You can click **Cancel** at any time to end the installation.
- 7 5. On the **Select the product you would like to install** page, click **Next**.
- 7 6. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup
7 wizard will guide you through the program setup process.
- 7 7. To proceed with the installation, you must accept the license agreement. On
7 the **License Agreement** page, select **I accept the terms in the license**
7 **agreement** and click **Next**.
- 7 8. Select **Install DB2 Information Center on this computer** on the **Select the**
7 **installation action** page. If you want to use a response file to install the DB2
7 Information Center on this or other computers at a later time, select **Save your**
7 **settings in a response file**. Click **Next**.
- 7 9. Select the languages in which the DB2 Information Center will be installed on
7 **Select the languages to install** page. Click **Next**.
- 7 10. Configure the DB2 Information Center for incoming communication on the
7 **Specify the DB2 Information Center port** page. Click **Next** to continue the
7 installation.
- 7 11. Review the installation choices you have made in the **Start copying files** page.
7 To change any settings, click **Back**. Click **Install** to copy the DB2 Information
7 Center files onto your computer.

7 You can install the DB2 Information Center using a response file. You can also use
7 the **db2rspgn** command to generate a response file based on an existing
7 installation.

7 For information on errors encountered during installation, see the db2.log and
7 db2wi.log files located in the 'My Documents'\DB2LOG\ directory. The location of
7 the 'My Documents' directory will depend on the settings on your computer.

7 The db2wi.log file captures the most recent DB2 installation information. The
7 db2.log captures the history of DB2 product installations.

- 7 **Related concepts:**
- 7 • “DB2 Information Center” on page 98
- 7 • “DB2 Information Center installation scenarios” on page 99
- 7 **Related tasks:**
- 7 • “Installing a DB2 product using a response file (Windows)” in the *Installation and Configuration Supplement*
- 7 • “Updating the DB2 Information Center installed on your computer or intranet server” on page 107
- 7 • “Displaying topics in your preferred language in the DB2 Information Center” on page 108
- 7 • “Invoking the DB2 Information Center” on page 106
- 7 • “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” on page 102
- 7 **Related reference:**
- 7 • “db2rspgn - Response File Generator Command (Windows)” in the *Command Reference*

Invoking the DB2 Information Center

7 The DB2 Information Center gives you access to all of the information that you
 7 need to use DB2 products for Linux, UNIX, and Windows operating systems such
 7 as DB2 Universal Database, DB2 Connect, DB2 Information Integrator, and DB2
 7 Query Patroller.

You can invoke the DB2 Information Center from one of the following places:

- Computers on which a DB2 UDB client or server is installed
- An intranet server or local computer on which the DB2 Information Center installed
- The IBM Web site

Prerequisites:

Before you invoke the DB2 Information Center:

- 7 • *Optional:* Configure your browser to display topics in your preferred language
- 7 • *Optional:* Configure your DB2 client to use the DB2 Information Center installed
- 7 on your computer or intranet server

Procedure:

To invoke the DB2 Information Center on a computer on which a DB2 UDB client or server is installed:

- From the Start Menu (Windows operating system): Click **Start** → **Programs** → **IBM DB2** → **Information** → **Information Center**.
- From the command line prompt:
 - For Linux and UNIX operating systems, issue the **db2icdocs** command.
 - For the Windows operating system, issue the **db2icdocs.exe** command.

To open the DB2 Information Center installed on an intranet server or local computer in a Web browser:

- Open the Web page at <http://<host-name>:<port-number>/>, where <host-name> represents the host name and <port-number> represents the port number on which the DB2 Information Center is available.

To open the DB2 Information Center on the IBM Web site in a Web browser:

- Open the Web page at publib.boulder.ibm.com/infocenter/db2help/.

Related concepts:

- “DB2 Information Center” on page 98
- “DB2 Information Center installation scenarios” on page 99

Related tasks:

- “Displaying topics in your preferred language in the DB2 Information Center” on page 108
- “Invoking contextual help from a DB2 tool” on page 115
- “Updating the DB2 Information Center installed on your computer or intranet server” on page 107
- “Invoking command help from the command line processor” on page 117
- “Setting the location for accessing the DB2 Information Center: Common GUI help”

Related reference:

- “HELP Command” in the *Command Reference*

Updating the DB2 Information Center installed on your computer or intranet server

The DB2 Information Center available from <http://publib.boulder.ibm.com/infocenter/db2help/> will be periodically updated with new or changed documentation. IBM may also make DB2 Information Center updates available to download and install on your computer or intranet server. Updating the DB2 Information Center does not update DB2 client or server products.

Prerequisites:

You must have access to a computer that is connected to the Internet.

Procedure:

To update the DB2 Information Center installed on your computer or intranet server:

1. Open the DB2 Information Center hosted on the IBM Web site at: <http://publib.boulder.ibm.com/infocenter/db2help/>
2. In the Downloads section of the welcome page under the Service and Support heading, click the **DB2 Universal Database documentation** link.
3. Determine if the version of your DB2 Information Center is out of date by comparing the latest refreshed documentation image level to the documentation level you have installed. The documentation level you have installed is listed on the DB2 Information Center welcome page.

4. If a more recent version of the DB2 Information Center is available, download the latest refreshed *DB2 Information Center* image applicable to your operating system.
5. To install the refreshed *DB2 Information Center* image, follow the instructions provided on the Web page.

Related concepts:

- “DB2 Information Center installation scenarios” on page 99

Related tasks:

- “Invoking the DB2 Information Center” on page 106
- “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” on page 102
- “Installing the DB2 Information Center using the DB2 Setup wizard (Windows)” on page 104

7 Displaying topics in your preferred language in the DB2 Information Center

7 The DB2 Information Center attempts to display topics in the language specified in
7 your browser preferences. If a topic has not been translated into your preferred
7 language, the DB2 Information Center displays the topic in English.

7 **Procedure:**

7 To display topics in your preferred language in the Internet Explorer browser:

- 7 1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...**
7 button. The Language Preferences window opens.
- 7 2. Ensure your preferred language is specified as the first entry in the list of
7 languages.
 - 7 • To add a new language to the list, click the **Add...** button.

7 **Note:** Adding a language does not guarantee that the computer has the fonts
7 required to display the topics in the preferred language.

- 7 • To move a language to the top of the list, select the language and click the
7 **Move Up** button until the language is first in the list of languages.
- 7 3. Refresh the page to display the DB2 Information Center in your preferred
7 language.

7 To display topics in your preferred language in the Mozilla browser:

- 7 1. In Mozilla, select the **Edit** —> **Preferences** —> **Languages** button. The
7 Languages panel is displayed in the Preferences window.
- 7 2. Ensure your preferred language is specified as the first entry in the list of
7 languages.
 - 7 • To add a new language to the list, click the **Add...** button to select a language
7 from the Add Languages window.
 - 7 • To move a language to the top of the list, select the language and click the
7 **Move Up** button until the language is first in the list of languages.
- 7 3. Refresh the page to display the DB2 Information Center in your preferred
7 language.

Table 2. Administration information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Administration Guide: Planning</i>	SC09-4822	db2d1x81
<i>IBM DB2 Universal Database Administration Guide: Implementation</i>	SC09-4820	db2d2x81
<i>IBM DB2 Universal Database Administration Guide: Performance</i>	SC09-4821	db2d3x81
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x81
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx81
<i>IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference</i>	SC09-4831	db2hax81
<i>IBM DB2 Universal Database Data Warehouse Center Administration Guide</i>	SC27-1123	db2ddx81
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1x81
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2x81
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x81

Application development information

The information in these books is of special interest to application developers or programmers working with DB2 Universal Database (DB2 UDB). You will find information about supported languages and compilers, as well as the documentation required to access DB2 UDB using the various supported programming interfaces, such as embedded SQL, ODBC, JDBC, SQLJ, and CLI. If you are using the DB2 Information Center, you can also access HTML versions of the source code for the sample programs.

Table 3. Application development information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Application Development Guide: Building and Running Applications</i>	SC09-4825	db2axx81
<i>IBM DB2 Universal Database Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1x81
<i>IBM DB2 Universal Database Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2x81

Table 3. Application development information (continued)

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db211x81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db212x81
<i>IBM DB2 Universal Database Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2adx81
<i>IBM DB2 XML Extender Administration and Programming</i>	SC27-1234	db2sxx81

Business intelligence information

The information in these books describes how to use components that enhance the data warehousing and analytical capabilities of DB2 Universal Database.

Table 4. Business intelligence information

Name	Form number	PDF file name
<i>IBM DB2 Warehouse Manager Standard Edition Information Catalog Center Administration Guide</i>	SC27-1125	db2dix81
<i>IBM DB2 Warehouse Manager Standard Edition Installation Guide</i>	GC27-1122	db2idx81
<i>IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager</i>	SC18-7727	iwhe1mstx80

DB2 Connect information

The information in this category describes how to access data on mainframe and midrange servers using DB2 Connect Enterprise Edition or DB2 Connect Personal Edition.

Table 5. DB2 Connect information

Name	Form number	PDF file name
<i>IBM Connectivity Supplement</i>	No form number	db2h1x81
<i>IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition</i>	GC09-4833	db2c6x81
<i>IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition</i>	GC09-4834	db2c1x81
<i>IBM DB2 Connect User's Guide</i>	SC09-4835	db2c0x81

Getting started information

The information in this category is useful when you are installing and configuring servers, clients, and other DB2 products.

Table 6. Getting started information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Clients</i>	GC09-4832, not available in hardcopy	db2itx81
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Servers</i>	GC09-4836	db2isx81
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Personal Edition</i>	GC09-4838	db2i1x81
<i>IBM DB2 Universal Database Installation and Configuration Supplement</i>	GC09-4837, not available in hardcopy	db2iyx81
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Data Links Manager</i>	GC09-4829	db2z6x81

Tutorial information

Tutorial information introduces DB2 features and teaches how to perform various tasks.

Table 7. Tutorial information

Name	Form number	PDF file name
<i>Business Intelligence Tutorial: Introduction to the Data Warehouse</i>	No form number	db2tux81
<i>Business Intelligence Tutorial: Extended Lessons in Data Warehousing</i>	No form number	db2tax81
<i>Information Catalog Center Tutorial</i>	No form number	db2aix81
<i>Video Central for e-business Tutorial</i>	No form number	db2twx81
<i>Visual Explain Tutorial</i>	No form number	db2tvx81

Optional component information

The information in this category describes how to work with optional DB2 components.

Table 8. Optional component information

Name	Form number	PDF file name
<i>IBM DB2 Cube Views Guide and Reference</i>	SC18-7298	db2aax81

Table 8. Optional component information (continued)

Name	Form number	PDF file name
IBM DB2 Query Patroller Guide: Installation, Administration and Usage Guide	GC09-7658	db2dwx81
IBM DB2 Spatial Extender and Geodetic Extender User's Guide and Reference	SC27-1226	db2sbx81
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x82
DB2 Net Search Extender Administration and User's Guide	SH12-6740	N/A

Note: HTML for this document is *not* installed from the HTML documentation CD.

Release notes

The release notes provide additional information specific to your product's release and FixPak level. The release notes also provide summaries of the documentation updates incorporated in each release, update, and FixPak.

Table 9. Release notes

Name	Form number	PDF file name
DB2 Release Notes	See note.	See note.
DB2 Installation Notes	Available on product CD-ROM only.	Not available.

Note: The Release Notes are available in:

- XHTML and Text format, on the product CDs
- PDF format, on the PDF Documentation CD

In addition the portions of the Release Notes that discuss *Known Problems and Workarounds* and *Incompatibilities Between Releases* also appear in the DB2 Information Center.

To view the Release Notes in text format on UNIX-based platforms, see the Release.Notes file. This file is located in the DB2DIR/Readme/%L directory, where %L represents the locale name and DB2DIR represents:

- For AIX operating systems: /usr/opt/db2_08_01
- For all other UNIX-based operating systems: /opt/IBM/db2/V8.1

Related concepts:

- "DB2 documentation and help" on page 97

Related tasks:

- "Printing DB2 books from PDF files" on page 114
- "Ordering printed DB2 books" on page 114
- "Invoking contextual help from a DB2 tool" on page 115

Printing DB2 books from PDF files

You can print DB2 books from the PDF files on the *DB2 PDF Documentation* CD. Using Adobe Acrobat Reader, you can print either the entire book or a specific range of pages.

Prerequisites:

Ensure that you have Adobe Acrobat Reader installed. If you need to install Adobe Acrobat Reader, it is available from the Adobe Web site at www.adobe.com

Procedure:

To print a DB2 book from a PDF file:

1. Insert the *DB2 PDF Documentation* CD. On UNIX operating systems, mount the DB2 PDF Documentation CD. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Open `index.htm`. The file opens in a browser window.
3. Click on the title of the PDF you want to see. The PDF will open in Acrobat Reader.
4. Select **File** → **Print** to print any portions of the book that you want.

Related concepts:

- “DB2 Information Center” on page 98

Related tasks:

- “Mounting the CD-ROM (AIX)” in the *Quick Beginnings for DB2 Servers*
- “Mounting the CD-ROM (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Mounting the CD-ROM (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Ordering printed DB2 books” on page 114
- “Mounting the CD-ROM (Solaris Operating Environment)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “DB2 PDF and printed documentation” on page 109

Ordering printed DB2 books

If you prefer to use hardcopy books, you can order them in one of three ways.

Procedure:

Printed books can be ordered in some countries or regions. Check the IBM Publications website for your country or region to see if this service is available in your country or region. When the publications are available for ordering, you can:

- Contact your IBM authorized dealer or marketing representative. To find a local IBM representative, check the IBM Worldwide Directory of Contacts at www.ibm.com/planetwide
- Phone 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

7
7
7

- Visit the IBM Publications Center at <http://www.ibm.com/shop/publications/order>. The ability to order books from the IBM Publications Center may not be available in all countries.

At the time the DB2 product becomes available, the printed books are the same as those that are available in PDF format on the *DB2 PDF Documentation CD*. Content in the printed books that appears in the *DB2 Information Center CD* is also the same. However, there is some additional content available in DB2 Information Center CD that does not appear anywhere in the PDF books (for example, SQL Administration routines and HTML samples). Not all books available on the DB2 PDF Documentation CD are available for ordering in hardcopy.

Note: The DB2 Information Center is updated more frequently than either the PDF or the hardcopy books; install documentation updates as they become available or refer to the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/> to get the most current information.

Related tasks:

- “Printing DB2 books from PDF files” on page 114

Related reference:

- “DB2 PDF and printed documentation” on page 109

Invoking contextual help from a DB2 tool

7
7
7
7
7
7
7
7
7
7
7

Contextual help provides information about the tasks or controls that are associated with a particular window, notebook, wizard, or advisor. Contextual help is available from DB2 administration and development tools that have graphical user interfaces. There are two types of contextual help:

- Help accessed through the **Help** button that is located on each window or notebook
- Infopops, which are pop-up information windows displayed when the mouse cursor is placed over a field or control, or when a field or control is selected in a window, notebook, wizard, or advisor and F1 is pressed.

The **Help** button gives you access to overview, prerequisite, and task information. The infopops describe the individual fields and controls.

Procedure:

To invoke contextual help:

- For window and notebook help, start one of the DB2 tools, then open any window or notebook. Click the **Help** button at the bottom right corner of the window or notebook to invoke the contextual help.

You can also access the contextual help from the **Help** menu item at the top of each of the DB2 tools centers.

Within wizards and advisors, click on the Task Overview link on the first page to view contextual help.

- For infopop help about individual controls on a window or notebook, click the control, then click **F1**. Pop-up information containing details about the control is displayed in a yellow window.

Note: To display infopops simply by holding the mouse cursor over a field or control, select the **Automatically display infopops** check box on the **Documentation** page of the Tool Settings notebook.

Similar to infopops, diagnosis pop-up information is another form of context-sensitive help; they contain data entry rules. Diagnosis pop-up information is displayed in a purple window that appears when data that is not valid or that is insufficient is entered. Diagnosis pop-up information can appear for:

- Compulsory fields.
- Fields whose data follows a precise format, such as a date field.

Related tasks:

- “Invoking the DB2 Information Center” on page 106
- “Invoking message help from the command line processor” on page 116
- “Invoking command help from the command line processor” on page 117
- “Invoking SQL state help from the command line processor” on page 117
- “Access to the DB2 Information Center: Concepts help”
- “How to use the DB2 UDB help: Common GUI help”
- “Setting the location for accessing the DB2 Information Center: Common GUI help”
- “Setting up access to DB2 contextual help and documentation: Common GUI help”

7 Invoking message help from the command line processor

Message help describes the cause of a message and describes any action you should take in response to the error.

Procedure:

To invoke message help, open the command line processor and enter:
? XXXnnnnn

where XXXnnnnn represents a valid message identifier.

For example, ? SQL30081 displays help about the SQL30081 message.

Related concepts:

- “Introduction to messages” in the *Message Reference Volume 1*

Related reference:

- “db2 - Command Line Processor Invocation Command” in the *Command Reference*

7 Invoking command help from the command line processor

7 Command help explains the syntax of commands in the command line processor.

7 Procedure:

7 To invoke command help, open the command line processor and enter:

7 ? *command*

7 where *command* represents a keyword or the entire command.

7 For example, ? catalog displays help for all of the CATALOG commands, while ?
7 catalog database displays help only for the CATALOG DATABASE command.

7 Related tasks:

- 7 • “Invoking contextual help from a DB2 tool” on page 115
- 7 • “Invoking the DB2 Information Center” on page 106
- 7 • “Invoking message help from the command line processor” on page 116
- 7 • “Invoking SQL state help from the command line processor” on page 117

7 Related reference:

- 7 • “db2 - Command Line Processor Invocation Command” in the *Command*
7 *Reference*

7 Invoking SQL state help from the command line processor

7 DB2 Universal Database returns an SQLSTATE value for conditions that could be
7 the result of an SQL statement. SQLSTATE help explains the meanings of SQL
7 states and SQL state class codes.

7 Procedure:

7 To invoke SQL state help, open the command line processor and enter:

7 ? *sqlstate* or ? *class code*

7 where *sqlstate* represents a valid five-digit SQL state and *class code* represents the
7 first two digits of the SQL state.

7 For example, ? 08003 displays help for the 08003 SQL state, and ? 08 displays help
7 for the 08 class code.

7 Related tasks:

- 7 • “Invoking the DB2 Information Center” on page 106
- 7 • “Invoking message help from the command line processor” on page 116
- 7 • “Invoking command help from the command line processor” on page 117

DB2 tutorials

The DB2® tutorials help you learn about various aspects of DB2 Universal Database. The tutorials provide lessons with step-by-step instructions in the areas of developing applications, tuning SQL query performance, working with data warehouses, managing metadata, and developing Web services using DB2.

Before you begin:

You can view the XHTML versions of the tutorials from the Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>.

Some tutorial lessons use sample data or code. See each tutorial for a description of any prerequisites for its specific tasks.

DB2 Universal Database tutorials:

Click on a tutorial title in the following list to view that tutorial.

Business Intelligence Tutorial: Introduction to the Data Warehouse Center

Perform introductory data warehousing tasks using the Data Warehouse Center.

Business Intelligence Tutorial: Extended Lessons in Data Warehousing

Perform advanced data warehousing tasks using the Data Warehouse Center.

Information Catalog Center Tutorial

Create and manage an information catalog to locate and use metadata using the Information Catalog Center.

Visual Explain Tutorial

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2® products.

DB2 documentation

Troubleshooting information can be found throughout the DB2 Information Center, as well as throughout the PDF books that make up the DB2 library. You can refer to the "Support and troubleshooting" branch of the DB2 Information Center navigation tree (in the left pane of your browser window) to see a complete listing of the DB2 troubleshooting documentation.

DB2 Technical Support Web site

Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs), FixPaks and the latest listing of internal DB2 error codes, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>

DB2 Problem Determination Tutorial Series

Refer to the DB2 Problem Determination Tutorial Series Web site to find information on how to quickly identify and resolve problems you might encounter while working with DB2 products. One tutorial introduces you to the DB2 problem determination facilities and tools available, and helps you decide when to use them. Other tutorials deal with related topics, such

as "Database Engine Problem Determination", "Performance Problem Determination", and "Application Problem Determination".

See the full set of DB2 problem determination tutorials on the DB2 Technical Support site at <http://www.ibm.com/software/data/support/pdm/db2tutorials.html>

Related concepts:

- "DB2 Information Center" on page 98
- "Introduction to problem determination - DB2 Technical Support tutorial" in the *Troubleshooting Guide*

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following list specifies the major accessibility features in DB2[®] Version 8 products:

- All DB2 functionality is available using the keyboard for navigation instead of the mouse. For more information, see "Keyboard input and navigation."
- You can customize the size and color of the fonts on DB2 user interfaces. For more information, see "Accessible display."
- DB2 products support accessibility applications that use the Java[™] Accessibility API. For more information, see "Compatibility with assistive technologies" on page 120.
- DB2 documentation is provided in an accessible format. For more information, see "Accessible documentation" on page 120.

Keyboard input and navigation

Keyboard input

You can operate the DB2 tools using only the keyboard. You can use keys or key combinations to perform operations that can also be done using a mouse. Standard operating system keystrokes are used for standard operating system operations.

For more information about using keys or key combinations to perform operations, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard navigation

You can navigate the DB2 tools user interface using keys or key combinations.

For more information about using keys or key combinations to navigate the DB2 Tools, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard focus

In UNIX[®] operating systems, the area of the active window where your keystrokes will have an effect is highlighted.

Accessible display

The DB2 tools have features that improve accessibility for users with low vision or other visual impairments. These accessibility enhancements include support for customizable font properties.

Font settings

You can select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

For more information about specifying font settings, see [Changing the fonts for menus and text: Common GUI help](#).

Non-dependence on color

You do not need to distinguish between colors in order to use any of the functions in this product.

Compatibility with assistive technologies

The DB2 tools interfaces support the Java Accessibility API, which enables you to use screen readers and other assistive technologies with DB2 products.

Accessible documentation

Documentation for DB2 is provided in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen-reader.

Related concepts:

- [“Dotted decimal syntax diagrams” on page 120](#)

Related tasks:

- [“Keyboard shortcuts and accelerators: Common GUI help”](#)
- [“Changing the fonts for menus and text: Common GUI help”](#)

7 Dotted decimal syntax diagrams

7 Syntax diagrams are provided in dotted decimal format for users accessing the
7 Information Center using a screen reader.

7 In dotted decimal format, each syntax element is written on a separate line. If two
7 or more syntax elements are always present together (or always absent together),
7 they can appear on the same line, because they can be considered as a single
7 compound syntax element.

7 Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To
7 hear these numbers correctly, make sure that your screen reader is set to read out
7 punctuation. All the syntax elements that have the same dotted decimal number
7 (for example, all the syntax elements that have the number 3.1) are mutually
7 exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you
7 know that your syntax can include either USERID or SYSTEMID, but not both.

7 The dotted decimal numbering level denotes the level of nesting. For example, if a
7 syntax element with dotted decimal number 3 is followed by a series of syntax
7 elements with dotted decimal number 3.1, all the syntax elements numbered 3.1
7 are subordinate to the syntax element numbered 3.

7 Certain words and symbols are used next to the dotted decimal numbers to add
7 information about the syntax elements. Occasionally, these words and symbols
7 might occur at the beginning of the element itself. For ease of identification, if the
7 word or symbol is a part of the syntax element, it is preceded by the backslash (\)
7 character. The * symbol can be used next to a dotted decimal number to indicate
7 that the syntax element repeats. For example, syntax element *FILE with dotted
7 decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that
7 syntax element FILE repeats. Format 3* * FILE indicates that syntax element *
7 FILE repeats.

7 Characters such as commas, which are used to separate a string of syntax
7 elements, are shown in the syntax just before the items they separate. These
7 characters can appear on the same line as each item, or on a separate line with the
7 same dotted decimal number as the relevant items. The line can also show another
7 symbol giving information about the syntax elements. For example, the lines 5.1*,
7 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the
7 LASTRUN and DELETE syntax elements, the elements must be separated by a
7 comma. If no separator is given, assume that you use a blank to separate each
7 syntax element.

7 If a syntax element is preceded by the % symbol, this indicates a reference that is
7 defined elsewhere. The string following the % symbol is the name of a syntax
7 fragment rather than a literal. For example, the line 2.1 %OP1 means that you
7 should refer to separate syntax fragment OP1.

7 The following words and symbols are used next to the dotted decimal numbers:

- 7 • ? means an optional syntax element. A dotted decimal number followed by the ?
7 symbol indicates that all the syntax elements with a corresponding dotted
7 decimal number, and any subordinate syntax elements, are optional. If there is
7 only one syntax element with a dotted decimal number, the ? symbol is
7 displayed on the same line as the syntax element, (for example 5? NOTIFY). If
7 there is more than one syntax element with a dotted decimal number, the ?
7 symbol is displayed on a line by itself, followed by the syntax elements that are
7 optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you
7 know that syntax elements NOTIFY and UPDATE are optional; that is, you can
7 choose one or none of them. The ? symbol is equivalent to a bypass line in a
7 railroad diagram.
- 7 • ! means a default syntax element. A dotted decimal number followed by the !
7 symbol and a syntax element indicates that the syntax element is the default
7 option for all syntax elements that share the same dotted decimal number. Only
7 one of the syntax elements that share the same dotted decimal number can
7 specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and
7 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword.
7 In this example, if you include the FILE keyword but do not specify an option,
7 default option KEEP will be applied. A default option also applies to the next
7 higher dotted decimal number. In this example, if the FILE keyword is omitted,
7 default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!
7 (KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next
7 higher dotted decimal number, 2.1 (which does not have an associated
7 keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE
7 is omitted.
- 7 • * means a syntax element that can be repeated 0 or more times. A dotted
7 decimal number followed by the * symbol indicates that this syntax element can
7 be used zero or more times; that is, it is optional and can be repeated. For
7 example, if you hear the line 5.1* data area, you know that you can include one

7 data area, more than one data area, or no data area. If you hear the lines 3*, 3
7 HOST, and 3 STATE, you know that you can include HOST, STATE, both
7 together, or nothing.

7 **Notes:**

- 7 1. If a dotted decimal number has an asterisk (*) next to it and there is only one
7 item with that dotted decimal number, you can repeat that same item more
7 than once.
- 7 2. If a dotted decimal number has an asterisk next to it and several items have
7 that dotted decimal number, you can use more than one item from the list,
7 but you cannot use the items more than once each. In the previous example,
7 you could write HOST STATE, but you could not write HOST HOST.
- 7 3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.
- 7 • + means a syntax element that must be included one or more times. A dotted
7 decimal number followed by the + symbol indicates that this syntax element
7 must be included one or more times; that is, it must be included at least once
7 and can be repeated. For example, if you hear the line 6.1+ data area, you must
7 include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE,
7 you know that you must include HOST, STATE, or both. Similar to the * symbol,
7 the + symbol can only repeat a particular item if it is the only item with that
7 dotted decimal number. The + symbol, like the * symbol, is equivalent to a
7 loop-back line in a railroad syntax diagram.

7 **Related concepts:**

- 7 • “Accessibility” on page 119

7 **Related tasks:**

- 7 • “Keyboard shortcuts and accelerators: Common GUI help”

7 **Related reference:**

- 7 • “How to read the syntax diagrams” in the *SQL Reference, Volume 2*

7 Common Criteria certification of DB2 Universal Database products

7 For Version 8.2, DB2 Universal Database (DB2 UDB) products are certified
7 according to the Common Criteria EAL4 (<http://niap.nist.gov/cc-scheme/>). The
7 following products are certified on the following operating systems:

7 *Table 10. Certified DB2 Universal Database configurations*

	Windows® 2000	Linux SuSE	AIX® 5.2	Solaris Operating Environment, 8
Enterprise Server Edition Note: Single-partition environment only.	Yes	Yes	Yes	Yes
Workgroup Server Edition	Yes	Yes	Yes	Yes
Personal Edition	Yes	Yes	N/A	N/A
Express Edition	Yes	Yes	N/A	N/A

7 **Notes:**

- 7 1. DB2® UDB configurations are Common Criteria certified on 32-bit hardware
7 only. 64-bit configurations are not certified.
- 7 2. DB2 UDB configurations on the Linux SuSE environment are Common Criteria
7 certified on Intel-based hardware only.

- 7
7
7
7
7
7
3. In a Common Criteria certified DB2 UDB environment, DB2 UDB clients are supported on the following operating systems:
- Windows 2000
 - Linux SuSE
 - AIX 5.2
 - Solaris Operating Environment, 8

7
7

For information about installing and configuring a DB2 UDB system that conforms to the Common Criteria EAL4, see the following books:

- 7
7
7
7
7
7
7
7
- *DB2 Universal Database Common Criteria Certification: Installing DB2 Universal Database Enterprise Server Edition and DB2 Universal Database Workgroup Server Edition*
 - *DB2 Universal Database Common Criteria Certification: Installing DB2 Universal Database Personal Edition*
 - *DB2 Universal Database Common Criteria Certification: Installing DB2 Universal Database Express Edition*
 - *DB2 Universal Database Common Criteria Certification: Administration and User Documentation*

7
7

These books are available in PDF format from the DB2 Information Management Library.

Notices

IBM may not offer the products, services, or features discussed in this document in all 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/region or send inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country/region 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.

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

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 that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extenders	SQL/400
DB2 OLAP Server	SQL/DS
DB2 Information Integrator	System/370
DB2 Query Patroller	System/390
DB2 Universal Database	SystemView
Distributed Relational Database Architecture	Tivoli
DRDA	VisualAge
eServer	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
IBM	WebSphere
IMS	WIN-OS/2
IMS/ESA	z/OS
	zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at <http://www.ibm.com/planetwide>

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at <http://www.ibm.com/software/data/db2/udb>

This site contains the latest information on the technical library, ordering books, product downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide



Printed in USA

Spine information:



IBM® DB2 Universal Database™

DB2 Glossary

Version 8.2