



IBM Informix Glossary



IBM Informix Glossary

Note:

Before using this information and the product it supports read the information in “Notices” on page B-1.

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.

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 2007, 2008. All rights reserved.

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

Introduction

About This Publication	iii
Types of Users	iii
Assumptions About Your Locale	iv
Documentation Conventions	iv
Typographical Conventions	iv
Feature, Product, and Platform Markup	iv
Example Code Conventions	v
Additional Documentation	v
Compliance with Industry Standards	vi
How to Provide Documentation Feedback	vi

About This Publication

This publication includes information about the system catalog tables, data types, and environment variables that IBM Informix products use.

This publication is one of a series of publications that discusses the Informix[®] implementation of SQL. The *IBM Informix Guide to SQL: Syntax* contains all the syntax descriptions for SQL and stored procedure language (SPL). The *IBM Informix Guide to SQL: Tutorial* shows how to use basic and advanced SQL and SPL routines to access and manipulate the data in your databases. The *IBM Informix Database Design and Implementation Guide* shows how to use SQL to implement and manage your databases.

See the documentation notes files for a list of the publications in the documentation set of your Informix database server.

Types of Users

This publication is written for the following users:

- Database users
- Database administrators
- Database server administrators
- Database-application programmers
- Performance engineers

This publication assumes that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming
- Some experience with database server administration, operating-system administration, or network administration

If you have limited experience with relational databases, SQL, or your operating system, refer to the *IBM Informix Dynamic Server Getting Started Guide* for your database server for a list of supplementary titles.

Assumptions About Your Locale

IBM Informix products can support many languages, cultures, and code sets. All the information related to character set, collation, and representation of numeric data, currency, date, and time is brought together in a single environment, called a Global Language Support (GLS) locale.

This publication assumes that your database uses the default locale. This default is **en_us.8859-1** (ISO 8859-1) on UNIX[®] platforms or **en_us.1252** (Microsoft[®] 1252) in Windows environments. This locale supports U.S. English format conventions for displaying and entering date, time, number, and currency values. It also supports the ISO 8859-1 (on UNIX and Linux[®]) or Microsoft 1252 (on Windows) code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or in SQL identifiers, or if you plan to use other collation rules for sorting character data, you need to specify the appropriate nondefault locale.

For instructions on how to specify a nondefault locale, and for additional syntax and other considerations related to GLS locales, see the *IBM Informix GLS User's Guide*.

Documentation Conventions

This section describes the following conventions, which are used in the product documentation for IBM[®] Informix Dynamic Server:

- Typographical conventions
- Feature, product, and platform conventions
- Example code conventions

Typographical Conventions

This publication uses the following conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	Keywords of SQL, SPL, and some other programming languages appear in uppercase letters in a serif font.
<i>italics</i>	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface	Names of program entities (such as classes, events, and tables), environment variables, file names, path names, and interface elements (such as icons, menu items, and buttons) appear in boldface.
monospace	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
>	This symbol indicates a menu item. For example, "Choose Tools > Options " means choose the Options item from the Tools menu.

Feature, Product, and Platform Markup

Feature, product, and platform markup identifies paragraphs that contain feature-specific, product-specific, or platform-specific information. Some examples

of this markup follow:

Dynamic Server
Identifies information that is specific to IBM Informix Dynamic Server
End of Dynamic Server

Windows Only
Identifies information that is specific to the Windows operating system
End of Windows Only

This markup can apply to one or more paragraphs within a section. When an entire section applies to a particular product or platform, this is noted as part of the heading text, for example:

Table Sorting (Windows)

Example Code Conventions

Examples of SQL code occur throughout this publication. Except as noted, the code is not specific to any single IBM Informix application development tool.

If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```
CONNECT TO stores_demo
...

DELETE FROM customer
  WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using an SQL API, you must use EXEC SQL at the start of each statement and a semicolon (or other appropriate delimiter) at the end of the statement. If you are using DB–Access, you must delimit multiple statements with semicolons.

Tip: Ellipsis points in a code example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the documentation for your product.

Additional Documentation

You can view, search, and print all of the product documentation from the IBM Informix Dynamic Server information center on the Web at <http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp>.

For additional documentation about IBM Informix Dynamic Server and related products, including release notes, machine notes, and documentation notes, go to the online product library page at <http://www.ibm.com/software/data/informix/>

pubs/library/. Alternatively, you can access or install the product documentation from the Quick Start CD that is shipped with the product.

Compliance with Industry Standards

The American National Standards Institute (ANSI) and the International Organization of Standardization (ISO) have jointly established a set of industry standards for the Structured Query Language (SQL). IBM Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992. In addition, many features of IBM Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL Common Applications Environment (CAE) standards.

How to Provide Documentation Feedback

You are encouraged to send your comments about IBM Informix user documentation by using one of the following methods:

- Send e-mail to docinf@us.ibm.com.
- Go to the Information Center at <http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp> and open the topic that you want to comment on. Click **Feedback** at the bottom of the page, fill out the form, and submit your feedback.

Feedback from both methods is monitored by those who maintain the user documentation of Dynamic Server. The feedback methods are reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact IBM Technical Support. For instructions, see the IBM Informix Technical Support Web site at <http://www.ibm.com/planetwide/>.

We appreciate your suggestions.

Terms

8-bit character. A single-byte character that consists of eight bits, which means that the code point is in the range 128 through 255. Examples from the ISO8859-1 code set include the non-English é, ñ, and ö characters. They can be interpreted correctly only by software that is *8-bit clean*.

8-bit clean. An attribute of software that can process character data that contains *8-bit characters*. The operating system or the database server reads the eighth bit as part of the code value. In other words, it does not ignore the eighth bit nor make its own interpretation of the eighth bit.

16-bit code set. Any *code set* that represents each character by two bytes, so that approximately 65,000 distinct characters can be encoded. Also called *double-byte code set*. An example is JIS X0208.

access method. (1) Any of a group of routines that access or manipulate a table or an index. In the output of a SET EXPLAIN statement, *access method* refers to the mode of table access in a query (for example, SEQUENTIAL SCAN as opposed to INDEX PATH). (2) A set of server routines that the database server uses to store and access the data in an index or a table. B-tree is the default secondary access method. Some DataBlade® modules have their own access methods, with routines defined by the module.

access privilege. The type of operation that a user has permission to perform in a specific database or database object. The Informix database, table, table-fragment, index, and column-access privileges are independent of operating-system access permissions.

active set. The collection of rows that satisfies a query associated with a cursor.

ActiveX value object. A Microsoft Common Object Model (COM)-compliant object that contains a client-side copy of an opaque type and its support routines. *See also* value object.

aggregate function. An SQL function that returns one value for a group of retrieved rows; for example, the frequency, sum, average, maximum, or minimum of an expression in a query or report. *See also* user-defined aggregate.

aggregate support function. One of a group of *user-defined functions* that the database server uses to calculate a user-defined aggregate.

AIO VP. *See* “asynchronous disk I/O virtual processor” on page 1-2.

alias. In an SQL query or in a form-specification file, an *alias* is a single-word temporary alternative name that is used in place of a qualified table name (for example, **t1** as an alias for *owner.table_name*). Aliases are often used in complex subqueries and are required for a self-join.

ANSI compliant. For databases, conforming to ANSI/ISO standards for the SQL language. An IBM Informix database can be created as either ANSI compliant or not ANSI compliant. An ANSI-compliant database enforces certain ANSI requirements, such as implicit transactions, explicit owner naming, no public synonyms, and unbuffered logging, which are not enforced in databases that are not ANSI compliant.

API. *See* application programming interface (API).

applet. A small application program that performs a specific task and is usually portable between operating systems. Often written in Java™, applets can be downloaded from the Internet and run in a Web browser.

application development tool. Software, such as IBM Informix 4GL, which you can use to create and maintain a database. The software allows a user to send instructions and data to (and receive information from) the database server.

application process. A unit to which resources and locks are allocated. An application process involves the running of one or more programs. *See also* database server process.

application-productivity tools. Software tools that are used to write application programs.

application program. A complete, self-contained program, such as an editor or electronic mail, that performs a specific task for the user, in contrast to system software, such as the operating system kernel, server processes, and program libraries.

application programming interface (API). An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program. *See also* SQL API and DataBlade API.

arbitrary rule. Expressions that define a fragmentation strategy. Unlike a *range rule*, an arbitrary rule can specify any relational and logical operators of SQL to define expressions (such as the OR operator to group data).

archiving. To copy all the data and indexes of a database to a different physical device from that which stores the database. Archived material can be used by the Database Administrator for recovery from a failure. *See also* backup.

arithmetic function. A function that returns a value by performing a mathematical operation on one or more arguments.

argument. A value that is passed to a *routine* or command. *Compare with* parameter.

array. A structure that contains an ordered group of data elements. All elements in an array have the same data type.

ASCII. American Standards Code for Information Interchange. A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters. *See also* single-byte character.

ASF. Associated Services Facility. The ASF code in IBM Informix products controls the connections between client applications and database servers.

asynchronous disk I/O virtual processor. A virtual processor that performs nonlogging disk I/O. *See also* "virtual processor" on page 1-28.

attached index. An index that has the same distribution scheme as its table. *See also* detached index, system index, and user index.

audit event. Any database server activity or operation that can potentially access or alter data. Audit events can be recorded and monitored by the database *secure auditing* facility. Examples of audit events include accessing tables, altering indexes, dropping chunks, granting database access, updating the current row, and running database utilities. (For a complete list of audit events, see the *IBM Informix: Trusted Facility Guide*.)

audit file. A file that contains records of audit events and resides in the specified audit directory. Audit files provide an audit trail of information that can be extracted by the database *secure auditing* facility for analysis.

audit mask. A structure that specifies which audit events should be logged (or excluded from logging) by the database *secure auditing facility*.

autocommit mode. A mode in which a COMMIT statement is automatically executed after each statement sent to the database server.

backup. The process of copying a file, directory, file system, or other data onto a tape, disk, or other media as insurance against data loss or corruption.

backup volume. Backup media, such as magnetic tape or optical disk.

base table. (1) A table referenced in the definition of a view, containing data that the view displays. A *multitable view* has more than one base table. (2) Within a table hierarchy of Dynamic Server, a *parent table* from whose schema another table (the *child table*) is derived. *See also* table hierarchy.

base type. *See* opaque data type.

bitmap index. A type of index that stores a bitmap for any highly duplicate key value. The bitmap indicates which rows have the duplicate key value. You create a bitmap index with the USING BITMAP keywords in the CREATE INDEX statement. *See also* B-tree index.

blobpage. The unit of disk allocation within a blobspace. The database server administrator determines the size of a blobpage. The size can vary, depending on the size of the TEXT or BYTE data that the user inserts.

blobspace. A logical collection of *chunks* that is used to store TEXT and BYTE data.

Boolean. Characteristic of an expression or variable that can only have a value of true or false.

Boolean function. A function that returns a Boolean value (true or false). A Boolean function can act as a *filter*.

bootstrap. A small program that loads larger programs during system initialization.

branch node. An index page that contains pointers to a leaf node or other branch nodes. The database server creates a branch node when the root node and subsequent leaf nodes become full.

branch page. A location on a tree structure that has at least one page below and one page above it. In an R-tree index, branch pages are located in the intermediate levels, between the root page and leaf pages.

B-tree. A model (with "simple graph" topology) for defining the logical structure of an index.

B-tree index. A type of index that uses a balanced tree structure for efficient record retrieval. A B-tree index is balanced when the leaf nodes are all at the same level from the root node. B-tree indexes store a list of rowids for any duplicate key value data in ascending or descending order. *See also* bitmap index and R-tree index.

buffer. A portion of computer memory where a program temporarily stores data. Data typically is read into or written from buffers to disk.

An area of storage that compensates for the different speeds of data flow or timings of events by temporarily holding a block of data that is waiting to be processed or written to an I/O device.

buffered disk I/O. Disk I/O that is controlled by the operating system instead of by an application. With buffered disk I/O, the operating system stores data in the kernel portion of memory before periodically writing the data to disk. *See also* unbuffered disk I/O and disk I/O.

buffered logging. A type of logging that holds transactions in a memory buffer until the buffer is full, regardless of when the transaction is committed or rolled back. Informix database servers provide this option to speed up operations by reducing the number of disk write operations.

built-in cast. A cast that is built into the database server. A built-in cast performs automatic conversions between different built-in data types.

built-in data type. A predefined data type that the database server supports; for example, INTEGER, CHAR, or SERIAL.

built-in function. A predefined, SQL-invoked function that provides some basic arithmetic and other operations, such as `cos()`, `log()`, or `today()`.

Cartesian product. The set that results when you pair each and every member of one set with each and every member of another set. A Cartesian product results from a multiple-table query when you do not specify the joining conditions among tables.

cascading deletes. Deleting rows from a child table when the foreign key is deleted from the parent table. When any rows are deleted from the primary key column of a table, cascading deletes, if enabled, delete identical information from any foreign-key column in a related table.

cast. A database object that converts one data type to another. Most built-in data types have built-in casts (that is, system-defined casts) to compatible data types. *See also* user-defined cast, explicit cast, and implicit cast.

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.

cast support function. A function that is used to implement an implicit or explicit cast by performing the necessary operations for conversion between two data types.

certificate authority. A trusted entity or organization that issues digital certificates for Secure Sockets Layer (SSL) communications.

character special device. *See* unbuffered disk I/O.

check constraint. A logical condition that must be satisfied before data values can be entered into a column of a database table during an INSERT or UPDATE statement.

checkpoint. A point during a database server operation when the pages on disk are synchronized with the pages in the shared memory buffer pool to allow the server to be restarted in case of interruption.

child table. The referencing table in a referential constraint. *See also* parent table.

class. A category of objects that have common properties and are managed through a specific system table. Informix database classes include access methods, aggregates, casts, routines, operators, tables, and types.

CLASSPATH. In the execution environment, a variable that specifies the directories in which to look for class and resource files.

client application. A program that requests services from a server program, typically a file server or a database server. For the GLS feature, the term *client application* includes database server utilities.

client computer. The computer on which a client application runs.

client files. The files that reside on a client workstation that accesses DataBlade module objects. Not all DataBlade modules have client files. Examples include client applications or client libraries that are specific to the DataBlade module.

client locale. The locale that a client application uses to perform read and write operations on the client computer. The `CLIENT_LOCALE` environment variable can specify a nondefault locale. *See also* server locale and locale.

client/server architecture. A hardware and software design that allows the user interface and database server to reside on separate nodes or platforms on a single computer or over a network. *See also* ASF, client application, and server-processing locale.

client/server connection statements. The SQL statements that can make connections with databases. These statements include `CONNECT`, `DISCONNECT`, and `SET CONNECTION`.

code set. The representation of a character set that specifies how to map each element of a character set to a unique code point. For example, ASCII, ISO8859-1, Microsoft 1252, and EBCDIC are code sets to represent the English language. A locale name specifies a code set.

code-set conversion. The process of converting character data from one code set (the *source* code set) to another (the *target* code set). Code-set conversion is useful when the client and server computers use different code sets to represent the same character data.

code-set order. The serialized order of characters within a code set. For example, in the ASCII code set, uppercase characters (A through Z) are ordered before lowercase characters (a through z). *See also* collation order and localized order.

collating sequence. *See* collation order, code-set order, and localized order.

collation. The process of character and string sorting based on alphabetical order and equivalence class.

collation order. The logical order in which the character-string values in a database are sorted and indexed. This is based on either the order of the code set or else some locale-specific order.

collection. An instance of a collection data type; a group of *elements* of the same *data type* stored in a SET, MULTISET, or LIST data type.

collection cursor. A database cursor that has an IBM Informix ESQL/C collection variable associated with it and provides access to the individual *elements* of a column whose data type is a *collection data type*.

collection data type. A complex data type whose instances are groups of *elements* of the same *data type*, which can be any *opaque data type*, *distinct data type*, *built-in data type*, *collection data type*, or *row data type*.

collection-derived table. A table that IBM Informix ESQL/C or SPL creates for a collection column when it encounters the TABLE keyword in an INSERT, DELETE, UPDATE, or SELECT statement. ESQL/C and SPL store this table in a collection variable to access *elements* of the collection as rows of the collection-derived table.

collection subquery. A query that takes the result of a subquery and turns it into an expression by using the MULTISSET keyword to convert returned values into a MULTISSET collection.

collection variable. An IBM Informix ESQL/C *host variable* or *SPL variable* that holds an entire *collection* and provides access, through a *collection cursor*, to the individual *elements* of the collection.

column expression. An expression that includes a column name and optionally uses *column subscripts* to define a column substring.

column distribution. See “data distribution” on page 1-6.

column statistics. See “data distribution” on page 1-6.

command file. A system file that contains one or more statements or commands, such as SQL statements.

commit work. To complete a transaction by accepting all changes to the database since the beginning of the transaction. See also roll back.

committed read. An Informix level of isolation in which the user can view only rows that are currently committed at the moment when the query is requested; the user cannot view rows that were changed as a part of a currently uncommitted transaction. Committed Read is available through a database server and set with the SET ISOLATION statement. It is the default level of isolation for databases that are not ANSI compliant. See also read committed.

commutator function. A Boolean function that accepts the same two arguments, in reverse order, as another Boolean function, and returns the same result. The query optimizer might choose the commutator function if it executes more quickly in a given query than the specified function.

compatible data types. Two data types for which casts exist in the database. See also implicit cast.

compile. To translate all or part of a program expressed in a high-level language into a computer program expressed in an intermediate language, an assembly language, or a machine language.

compile-time error. An error that occurs when you *compile* the program source code. This type of error indicates syntax errors in the source code. Compare with runtime error.

complex data type. A *data type* that is built from a combination of other data types by using an SQL type constructor and whose components can be accessed through SQL statements. See also row data type and collection data type.

complex qualification. A WHERE clause in a query in which two or more logical operators are used on the same column on which the R-tree index is defined.

composite data type. See row data type.

composite index. An index constructed on two or more columns of a table. The order imposed by the composite index varies least frequently on the first-named column and most frequently on the last-named column.

compressed bitmap. An indexing method that identifies records through a fragment identifier and a record identifier.

concatenation operator. An operator whose notation is composed of two pipe symbols (||); this is used in expressions to indicate the joining of two strings.

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

configuration file. A file read during database server disk or shared-memory initialization that contains the parameters that specify values for configurable behavior. A database server and its archiving tool use configuration files.

constraint. A rule that limits the values that can be inserted, deleted, or updated in a table.

constructed data type. A complex data type created with a type constructor; for example, a collection data type or an unnamed row data type.

constructor. *See* type constructor.

cooked files. *See* buffered disk I/O.

coordinating server. In a query that spans multiple database servers, the server in which the query is initiated is called the *coordinator* or *coordinating server*. This server is also sometimes called the *local server* because it is the local server to the client initiating the query. To respond to the query, the coordinating server starts sessions on the other servers involved in the query. *See also* distributed query and subordinate server.

correlated subquery. A subquery (or inner SELECT) that depends on a value produced by the outer SELECT statement that contains it. Also a nested subquery whose WHERE clause refers to an attribute of a relation that is declared in an outer SELECT. Correlated subqueries reference one or more columns from one or more tables of a parent query and need to be evaluated once for each row in the parent query. *See also* subquery.

correlation name. The prefix used with a column name in a triggered action to refer to an old (before triggering statement) or a new (after triggering statement) column value. The associated column must be in the triggering table. *See also* trigger.

corrupted database. A database whose tables or indexes contain incomplete, inconsistent, or incorrect data.

corrupted index. An index that does not correspond exactly to the data in its table.

cross-server query. *See* distributed query.

current row. The most recently retrieved row of the active set of a query.

cursor. In SQL, an identifier associated with a group of rows or with a collection data type. Conceptually, the pointer to the current row or collection element. You can use cursors for SELECT statements or EXECUTE PROCEDURE statements (associating the cursor with the rows returned by a query) or INSERT statements (associating the cursor with a buffer to insert multiple rows as a group).

cursor function. A user-defined routine (UDR) that returns one or more rows of data and therefore requires a cursor to execute. An *SPL routine* is a cursor function when its RETURN statement contains the WITH RESUME keywords. An *external function* is a cursor function when it is defined as an iterator function. *Compare with* noncursor function.

cursor manipulation statements. The SQL statements that control cursors; specifically, the CLOSE, DECLARE, FETCH, FLUSH, OPEN, and PUT statements.

cursor stability. 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.

custom installation. An installation option in which users control what products and features are installed and customize the installation to match their needs. This option is particularly useful for those who want to deploy applications that embed IBM IDS because it lets them control the amount of required disk space. *See also* "typical installation" on page 1-26.

data access statements. The subset of SQL statements that you can use to grant and revoke permissions and to lock tables.

data definition statements. The subset of SQL statements (sometimes called *data definition language*, or DDL) to create, alter, drop, and rename data objects, including databases, tables, views, synonyms, triggers, sequences, and user-defined routines.

data dictionary. The set of tables that keeps track of the structure of the database and the inventory of database objects. A data dictionary is also called the system catalog. Each database that a database server supports has its own system catalog.

data distribution. A mapping of the data values within a column into a set of categories that are equivalent to a histogram or to a frequency distribution. The query optimizer can use this information when it creates a plan for a query.

data file. A flat file containing data to be loaded into the database.

data integrity. The condition that exists as long as accidental or intentional destruction, alteration, or loss of data does not occur.

data integrity statements. SQL statements that you use to control transactions and audits. Data integrity statements also include statements for repairing and recovering tables.

data manipulation statements. The SQL statements that can query tables, insert into tables, delete from tables, or update tables (select, insert, delete, update). The load and unload utilities also are sometimes called data manipulation statements.

data object. The data that is stored in an R-tree indexed column of a table and in the R-tree index itself.

data partitioning. *See* table fragmentation.

data restriction. Synonym for *constraint*.

database administrator (DBA).

A person who is responsible for the design, development, operation, security, maintenance, and use of a database. Contrast with database server administrator (DBSA).

database application. A program that applies database management techniques to implement specific data manipulation and reporting tasks.

database environment. Used in the CONNECT statement. Either the database server or the database server and database to which a user connects.

database locale. The locale that defines the code set, collation, and date, time, number, and currency display conventions of a database server. The **DB_LOCALE** environment variable can specify this locale.

database management system. *See* DBMS.

database object. 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 security administrator . (1) A person who has the authority to implement and maintain label-based access control. (2) An IDS user with DBSECADM authority.

database server. A software package that manages access to one or more databases for one or more client applications. *See also* relational database server.

database server administrator (DBSA). A person who is responsible for managing one or more database servers.

database server process. A virtual processor that functions similarly to a CPU in a computer. *See also* application process.

database server utility. A program that performs a specific task. For example, DB-Access, **dbexport**, and **onmode** are Informix database server utilities.

database URL. A uniform resource locator (URL) that is passed to the **DriverManager.getConnection()** method that specifies the subprotocol (the database connectivity mechanism), the database or database server identifier, and a list of properties that can include Informix environment variables.

DataBlade API. An application programming interface (API) that allows a C *user-defined routine* access to the client application.

DataBlade API data types. A set of Informix C data types that correspond to some of the Informix SQL data types, including extended data types. Use these data types instead of the standard C data types to ensure portable applications.

DataBlade module. A group of *database objects* and supporting code that extends an object-relational database to manage new kinds of data or add new features. A DataBlade module can include new data types, *routines*, *casts*, *aggregates*, *access methods*, SQL code, client code, and installation programs.

data type. A descriptor assigned to each column in a table or program variable, which indicates the type of data the column or program variable is intended to hold. Informix data types for Global Language Support are discussed in the *IBM Informix: GLS User's Guide*. *See also* built-in data type, complex data type, distinct data type, opaque data type, and user-defined data type.

DBA. *Database Administrator.* A level of privilege, typically for operations that most users are not authorized to perform.

DBA-privileged. A class of SPL routines that only a user with DBA database privileges creates.

DBMS. *database management system.* Consists of all of the components that are necessary to create and maintain a database, including the application development tools and the database server.

DBSA. Database Server Administrator.

dbspace. A logical collection of one or more chunks. Because chunks represent specific regions of disk space, the creators of databases and tables can control where their data is physically located by placing databases or tables in specific dbspaces. A dbspace provides a link between logical (such as tables) and physical units (such as chunks) of storage. *See also* root dbspace.

DDL. Data definition language, A subset of the Structured Query Language (SQL) for declaring, modifying, and dropping database objects (such as tables, constraints, or indexes). *See also* data definition statement.

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 in use by the original transaction.

debug file. A file that receives output used for debugging purposes.

decision-support application. An application that provides information that is used for strategic planning, decision-making, and reporting. It typically executes in a batch environment in a sequential scan fashion and returns a large fraction of the rows scanned. Decision-support queries typically scan the entire database. *See also* online transaction processing application.

decision-support query. A query that a decision-support application generates. A decision support query often requires multiple joins, temporary tables, and extensive calculations, and can benefit significantly from PDQ. *See also* online transaction processing application.

declaration statement. A programming language statement that describes or defines objects; for example, defining a program variable. *Compare with* procedure. *See also* data definition statement.

delimited identifier. If the `DELIMIT` environment variable is set, this is an SQL identifier enclosed between double (") quotation marks. This supports identifiers that are SQL-reserved keywords or that contain whitespace or other characters outside the default SQL character set for identifiers.

delimiter. A flag that is formed by a character or a sequence of characters in order to group or separate items of data by marking the beginning and end of a unit of data. The delimiter is not a part of the flagged unit of data.

deployment wizard. *See* "custom installation" on page 1-6.

derived table. The set of rows returned by a subquery within a DML statement of SQL. Derived tables can be defined by simple, UNION, nested, or joined subqueries, including OUTER joins.

descriptor. A quoted string or variable that identifies an allocated system-descriptor area or an SQLDA structure. It is used for the Informix SQL APIs. *See also* identifier.

detached index. In a table with indexes, an index that uses a fragmentation strategy that is different from the table fragmentation.

device array. A list of I/O devices.

diagnostic area. A data structure (sometimes called SQLDA) that stores diagnostic information about an executed SQL statement.

diagnostics table. A special table that holds information about the integrity violations caused by each row in a violations table. You use the START VIOLATIONS TABLE statement to create violations tables and diagnostics tables and associate them with a base table.

digital certificate. An electronic ID card issued by a trusted party. This certificate provides assurance of the identity of a user or server for Secure Sockets Layer (SSL) communications.

dirty read. A read request that does not involve any locking mechanism, and which may obtain invalid data - that is, data that has been updated, but is not yet committed, by another task. This could also apply to data that is about to be updated, and which will be invalid by the time the reading task has completed.

disabled mode. The object mode in which a database object is disabled. When a constraint, index, or trigger is in the disabled mode, the database server acts as if the object does not exist and does not take it into consideration during the execution of data manipulation statements.

disk configuration. The organization of data on a disk; also refers to the process of preparing a disk to store data.

disk I/O. Fixed-disk input and output.

display label. A temporary name for a column or an expression in a query.

distinct data type. A *data type* that you declare with the CREATE DISTINCT TYPE statement. A distinct data type has the same internal storage representation as its *source type* (an existing *opaque data type*, *built-in data type*, *named row type*, or *distinct data type*) but a different name, and can have different casts and routines. To compare a distinct data type with its source type requires an *explicit cast*. A distinct data type inherits all routines that are defined on its source type.

distributed query. A query that accesses data from a database other than the current database.

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

distribution scheme. *See* table fragmentation.

DLL. dynamic link library (DLL).

DML. Data manipulation language. *See also* data manipulation statements.

document object model (DOM). A system in which a structured document, for example an XML file, is viewed as a tree of objects that can be programmatically accessed and updated.

dominant table.

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.

DRDA. Distributed Relational Database Architecture

DSS. Decision Support System. *See also* decision-support application.

duplicate index. An index that allows duplicate values in the indexed column.

dynamic link library (DLL). A *shared-object file* on a Windows® system. *See also* shared library.

dynamic management statements. The SQL statements that describe, execute, and prepare other statements.

dynamic routine-name specification. The execution of a *user-defined routine* whose name is determined at runtime through an *SPL variable* in the EXECUTE PROCEDURE, EXECUTE ROUTINE, or EXECUTE FUNCTION statement.

Dynamic Server instance. The set of processes, storage spaces, and shared memory that together comprise a complete database server. A single Dynamic Server instance can support more than one database.

dynamic SQL. An SQL statement that is prepared and executed at run time. In dynamic SQL, the SQL statement is contained as a character string in a host variable and is not precompiled.

dynamic statements. See dynamic SQL.

element. A member of a *collection*, such as a LIST, MULTISSET, or SET data type. An element can be a value of any *built-in data type, opaque data type, distinct data type, named row type, unnamed row type, or collection data type*.

enabled mode. The default object mode of database objects. When a constraint, index, or trigger is in this mode, the database server recognizes the existence of the object and takes the object into consideration while executing data manipulation statements.

end-user format. The format in which data appears within a client application as literal strings or character variables. End-user formats are useful for data types whose database format is different from the format to which users are accustomed.

end-user routine. A *user-defined routine* (UDR) that performs a task within an SQL statement that the existing *built-in* routines do not perform. Examples of tasks include encapsulating multiple SQL statements, creating trigger actions, and restricting who can access *database objects*.

environment variable. A variable that is included in the current software environment and is therefore available to any called program that requests it.

error log. A data set or file that is used to record error information about a product or system.

error message. A message that is associated with a (usually negative) number. IBM Informix applications display error messages on the screen or write them to files.

error trapping. See exception handling.

escape character. A character that indicates that the next character, normally interpreted by the program as having special significance, is to be processed as a literal character instead. The escape character precedes the special character (such as a wildcard or delimiter) to “escape” (that is, ignore) the special significance.

exception. Any user, logic, or system error detected by a function that does not itself deal with the error but passes the error on to a handling routine (also called throwing the exception).

exception handling. The performance of a specified response to an abnormal condition. Exception handling allows control and information to be passed to an exception handler when an exception occurs. In C++, try blocks, catch blocks, and throw expressions are the constructs used to implement formal exception handling.

exclusive access. Sole access to a database or table by a user. Other users are prevented from using the database or table.

exclusive lock. A lock that prevents concurrently executing application processes from accessing database data.

executable file. A file that contains code that can be executed directly. A C-language object file can be an executable file; it contains the machine-level instructions that correspond to the C-language *source file*.

exemption. Pertains to immunity from one or more rules of label-based access control. See also “LBAC credentials” on page 1-14.

explicit cast. A *user-defined cast* that a user explicitly invokes with the CAST AS keyword or cast operator (::). See also implicit cast.

explicit connection. A connection made to a database environment that uses the CONNECT statement. See also implicit connection.

explicit transaction. A transaction that is initiated by the BEGIN WORK statement. This type of transaction is available only in non-ANSI compliant databases that support logging. See also implicit transaction and singleton implicit transaction.

expression-based fragmentation. A method of partitioning a table or index into fragments in which the result of an expression determines the fragment in which a row will reside. You fragment tables to logically distribute data and thereby improve performance of queries that use the expression in their WHERE clause.

extended data type. A term used to refer to data types that are not built in; namely *complex data types, opaque data types, and distinct data types*.

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

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.

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.

external routine. A *user-defined routine* that is written in an external language that the database supports. These external languages include the C and Java languages. The routine names, parameters, and other information are registered in the system catalog tables of a database. However, the executable code of an external routine is stored outside of the database. An external routine can be an *external function* or an *external procedure*.

external space. Storage space that a user-defined access method manages rather than the database server. The IN clause of the CREATE TABLE and CREATE INDEX statements can specify the name of an external space instead of a dbspace.

external table. A database table that is not in the current database. It might or might not be in a database that the same database server manages.

extspace. A logical name associated with an arbitrary string that signifies the location of external data. You can access its contents with a user-defined access method.

fault tolerance. See high availability.

feature. Part of a product that is optional and that you can choose to install or not install. See also “custom installation” on page 1-6.

fetch. The action of moving a cursor to a new row and retrieving the row values into memory.

fetch buffer. A buffer in the application process that the database server uses to send fetched row data (except TEXT and BYTE data) to the application.

field. In a record, a specified area used for a particular category of data. For example, a record about an employee might be subdivided into fields containing the employee’s name, address, and salary.

FIFO. See “first-in-first-out (FIFO).”

fillfactor. An index parameter that specifies the percentage of an R-tree index page that should be filled with entries when the R-tree access method creates an R-tree index using the bottom-up build method.

filter. A set of conditions (sometimes called a *predicate*) for selecting rows or records. In an SQL query, the conditional expression in the WHERE clause is a filter that controls the active set of the query. The High-Performance Loader (HPL) uses a filter component to screen data before loading it into a database.

filtering mode. An object mode of a database object, causing bad rows to be written to the violations table during DML operations. During DML operations, the database server enforces requirements of a constraint or of a unique index that is in this mode and identifies any rows that would violate the requirement.

first-in-first-out (FIFO) . A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. See also “last-in first-out (LIFO)” on page 1-14.

fixed-point number. A number where the decimal point is fixed at a specific place regardless of the value of the number.

flexible temporary table. An explicit temporary table that Extended Parallel Server automatically fragments using a round-robin distribution scheme.

FLRU. See “free least-recently used (FLRU)” on page 1-12.

formatting character. For Extended Parallel Server, a percent sign (%) followed by a letter (c, n, o, or r). In a command line, Extended Parallel Server expands the formatting character to designate multiple coserver numbers (%c), multiple nodes (%n), multiple ordinal numbers designating dbspaces (%d), or a range of dbspaces (%r).

fragment. See table fragment.

fragmentation. A database server feature that allows the user to control where data is stored at the table level. Fragmentation enables the user to define groups of rows or index keys within a table according to some algorithm or scheme.

fragment elimination. The process of applying a filter predicate to the fragmentation strategy of a table or index and removing the fragments that do not apply to the operation.

free least-recently used (FLRU). A list that tracks free or unmodified pages in a queue.

function. A routine that returns one or more values. See also user-defined function.

functional index. An index that stores the result of executing a specified function on a table column.

function cursor. A cursor that is associated with an EXECUTE FUNCTION statement, which executes routines that return values. See also cursor function.

function overloading. See routine overloading.

fundamental data type. A data type that cannot be broken into smaller pieces by the database server using SQL statements; for example, built-in data types and opaque data types.

global catalog. A table that contains a global inventory of Enterprise Replication configuration information.

Global Language Support (GLS). A feature that enables Informix APIs and database servers to support different languages, cultural conventions, and code sets. For information about the GLS feature, see the *IBM Informix: GLS User's Guide*.

global variable. A variable or identifier whose scope of reference is all modules of a program. Compare with local variable.

GLS. Global Language Support

hash rule. A user-defined algorithm that maps each row in a table to a set of hash values and that is used to determine the fragment in which a row is stored.

HDR. See High-availability data replication.

heterogeneous commit. A protocol governing a group of database servers, of which at least one is a *gateway* participant. A heterogeneous commit ensures the all-or-nothing basis of distributed transactions in a heterogeneous environment.

hierarchy. The tree-like arrangement of segments in a database, beginning with the root segment and proceeding down to dependent segments.

high availability. The ability of a system to resist failure and data loss. High availability includes features such as fast recovery and mirroring. It is sometimes referred to as *fault tolerance*.

high availability cluster. A high-availability configuration consisting of a primary server and one or more secondary servers. Multiple types of secondary servers can coexist in a high-availability configuration. A secondary server can be an SD secondary server, an RS secondary server, or an HDR secondary server.

High-availability data replication. Copies data to a database server on another computer. You can also use ON-Bar with HDR to restore data on another computer.

High-Performance Loader. The High-Performance Loader (HPL) utility is part of Dynamic Server. The HPL loads and unloads data using parallel access to devices. See also external table.

hold cursor. A cursor that is created using the WITH HOLD keywords. A hold cursor remains open past the end of a transaction. It allows uninterrupted access to a set of rows across multiple transactions.

host variable. An SQL API program variable that you use in an embedded statement to transfer information between the SQL API program and the database.

IANA. Acronym for Internet Assigned Numbers Authority, which defines a hierarchy for naming tables and columns and for deriving numerical object identifiers (OIDs). IANA assigns identifiers to companies that use the SNMP protocol.

identifier. In the default locale, a sequence of letters, digits, and underscores (_) that is the unqualified name of a database, storage, or program object. (Additional characters are valid in other locales or if the **DELIMITED** variable is set.)

index page logging. Pertains to writing newly created index files to the logical log for the purpose of synchronizing index creation between servers in high-availability environments.

implicit cast. A *built-in* or *user-defined cast* that the database server automatically invokes to perform data-type conversion. *See also* explicit cast.

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

implicit transaction. A transaction that begins implicitly after the **COMMIT WORK** or **ROLLBACK WORK** statement. This is the only type of transaction that ANSI-compliant databases support, but it is also available for other databases that support logging. *See also* explicit transaction and singleton implicit transaction.

imported restore. Allows for the transfer of data from one instance of a database server to the same instance on a foreign host. In addition to completing the imported restore process, the imported restore is also a subset of these three processes: disaster recovery, database server upgrade, and initialization of HDR.

index self-join. A type of index scan that is a union of many small index scans, each one with a single unique combination of lead-key columns and filters on non-lead-key columns. The union of small index scans results in an access path that uses only subsets of the full range of a composite index.

INFORMIXDIR. The Informix environment variable that specifies the directory in which IBM Informix products are installed.

initialize. To prepare a system, device, or a program for operation.

input parameter. In a prepared SQL statement, a value, represented by a “?” placeholder symbol, that must be provided when the prepared statement is executed.

insert cursor. The position of the cursor marking where new characters will be added when entering text.

internationalization. In software engineering, the process of producing a product that is independent of any particular language, script, culture, and coded character set.

interquery parallelization. The ability to process multiple queries simultaneously to avoid a performance delay when multiple independent queries access the same table. *See also* intraquery parallelization.

interrupt key. A key used to cancel or abort a program or to leave a current menu and return to the menu one level above. On many systems, the interrupt key is **CONTROL-C**; on some others, the interrupt key is **DEL** or **CONTROL-Break**.

intraquery parallelization. Breaking of a single query into subqueries by a database server using PDQ and then processing the subqueries in parallel. Parallel processing of this type has important implications when each subquery retrieves data from a fragment of a table. Because each partial query operates on a smaller amount of data, the retrieval time is significantly reduced and performance is improved. *See also* interquery parallelization.

ISAM. *Indexed Sequential Access Method.* ISAM allows you to find information in a specific order or to find specific items of information quickly through an index. *See also* access method.

ISAM error. Operating system or file access error.

isolation level. When multiple users attempt to access common data, the level of independence specifically relating to the locking strategy for read-only SQL requests. The various levels of isolation are distinguished primarily by the length of time that shared locks are (or can be) acquired and held. You can set the isolation level with the **SET ISOLATION** or **SET TRANSACTION** statement.

iterator function. A cursor function that returns a data set by successive calls. It can be written in C or Java or can be an SPL routine that includes **RETURN WITH RESUME**.

jagged rows. A query result in which rows differ in the number and type of columns that they contain because the query applies to more than one table in a table hierarchy.

kernel. The part of an operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

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.

keystore. A Secure Sockets Layer (SSL) protected database that stores keys and digital certificates.

label-based access control (LBAC) . A security mechanism that uses security labels to restrict user access to individual table rows and columns. *See also* "security label" on page 1-21, "security policy" on page 1-21, "security label component" on page 1-21.

large object. A data object that is logically stored in a column of a table, but physically stored independently of the column, due to its size. Large objects can be *simple large objects* (TEXT, BYTE) or *smart large objects* (BLOB, CLOB).

last-in first-out (LIFO). A queuing technique in which the next item to be retrieved is the item most recently placed on the queue. *See also* "first-in-first-out (FIFO)" on page 1-11.

launchpad. A graphical interface for launching the product installation.

LBAC. *See* "label-based access control (LBAC)."

LBAC credentials. LBAC credentials are any security labels a user holds plus any exemptions that the user holds.

leaf node. Index page containing index items and horizontal pointers to other leaf nodes. A database server creates leaf nodes when the root node becomes full.

leaf page. A location on a tree structure that has at least one page above it and no pages below it. In an R-tree index, leaf pages are located in the final levels and contain data objects and row IDs.

least recently used (LRU) . A policy for a caching algorithm that removes from cache the item that has the longest elapsed time since its last access. An algorithm used to identify and make available the cache space that contains the data that was least recently used.

level of isolation. *See* isolation level.

library. A group of precompiled *routines* designed to perform tasks that are common to a given kind of application. An application programming interface (API) can include a library of routines that you can call from application programs. *See also* dynamic link library (DLL), shared library, and shared-object file.

light append. An unbuffered, unlogged insert operation.

link. A way of combining separately compiled program modules, usually into an executable program.

LIST constructor. A type constructor used to create a LIST data type.

LIST data type. A *collection data type* created with the LIST constructor in which *elements* are ordered and duplicates are allowed.

load job. The information required to load data into a relational database using the HPL. This information includes the format, map, filter, device array, project, and special options.

locale. The part of a user's environment that brings together information about how to handle data that is specific to the end user's particular country, language, or territory. The locale is typically specified when configuring the operating system or internationalized software products.

localization. *See* internationalization.

localized order. A collation order other than code-set order, if defined for a locale. Only NCHAR and NVARCHAR data values are collated in a localized order. Database objects collate in their creation-time order, if this is not the runtime order.

local loopback. A connection between the client application and database server that uses a network connection even though the client application and the database server are on the same computer.

local variable. A symbol defined in one program module or procedure that can only be used within that program module or procedure. *Compare with* global variable.

locking. A concurrency feature temporarily limiting access to an object (database, table, page, or row) to prevent conflicting interactions among concurrent processes. An exclusive lock restricts read and write access to only one user; a shared lock allows read-only access to other users. An update lock begins as a shared lock, but is upgraded to an exclusive lock after a row is changed.

locking granularity. The level and type of information that a lock protects.

lock mode. A representation for the type of access that concurrently running programs can have to a resource that a lock is holding.

logical log. An allocation of disk space that the database server manages that contains records of all changes that were performed on a database during the period when the log was active. The logical log is used to roll back transactions, recover from system failures, and restore databases from archives. *See also* physical log.

login. Refers to the process of signing on to a given computer system by typing in one's user ID and password.

loop label. In SPL routines, an SQL identifier whose declaration immediately precedes an iterative statement, and whose name immediately follows the terminating END FOR, END LOOP, or END WHILE keywords. The EXIT label statement can transfer program control to the first executable statement that follows the last statement in the FOR, FOR LOOP, LOOP, WHILE LOOP, or WHILE loop that has the specified loop label.

LRU. *See* "least recently used (LRU)" on page 1-14

LVARCHAR. A *built-in data type* that stores varying-length character data of up to 32 kilobytes.

MACH11 . *See* high availability cluster.

manifest file. An ASCII file, which must not be customized, that is automatically created during installation of an IDS instance. The file contains a history of installation activity. *Contrast with* "response file" on page 1-19.

master replicate. An Enterprise Replication replicate that guarantees data integrity by verifying that replicated tables on different servers have consistent column attributes. *See also* "replicate" on page 1-19, "shadow replicate" on page 1-22.

member. A component of an opaque data type. A member has a name and a data type and can be accessed in an SQL statement by user-defined accessor functions.

Memory Grant Manager (MGM). A database server component that coordinates the use of memory and I/O bandwidth for decision-support queries. MGM uses the DS_MAX_QUERIES, DS_TOTAL_MEMORY, DS_MAX_SCANS, and PDQPRIORITY configuration parameters to determine what resources can or cannot be granted to a decision-support query.

mirroring. Storing the same data on two chunks simultaneously. If one chunk fails, the data values are still usable on the other chunk in the mirrored pair. The database server administrator handles this data storage option.

MLRU . *See* "modified least-recently used (MLRU)."

modified least-recently used (MLRU). A list that tracks modified pages in a queue.

multibyte character. A character that might require from two to four bytes of storage. If a language contains more than 256 characters, the code set must contain multibyte characters. With a multibyte code set, an application cannot assume that one character requires only one byte of storage. *See also* single-byte character.

multiplexed connection. A single network connection between a database server and a client application that handles multiple database connections from the client.

MULTISET constructor. A type constructor used to create a MULTISET data type.

MULTISET data type. A *collection data type* created with the MULTISET constructor in which *elements* are not ordered and duplicates are allowed.

multithreading. A mode of operation in which the operating system can run different parts of a program, called threads, simultaneously. *See* thread.

named row data type. A *row data type* created with the CREATE ROW TYPE statement that has a declared name and inheritance properties and can be used to construct a *typed table*.

national character. In National Language Support (NLS), a character from the written form of a natural language that can be stored in an NCHAR or NVARCHAR column. Sometimes called a *native character*.

native character. *See* national character.

National Language Support (NLS). *See* Global Language Support (GLS).

node. (1) In the context of an index for a database, a node is an ordered group of key values having a fixed number of elements. (A key is a value from a data record.) A B-tree for example, is a set of nodes that contain keys and pointers arranged in a hierarchy. *See also* branch node, leaf node, and root node. (2) In hardware, a uniprocessor or symmetric multiprocessor (SMP) computer that is part of a clustered system or a massively parallel processing (MPP) system. *See also* symmetric multiprocessing system.

noncursor function. A *user-defined function* that returns a single group of values (one row of data) and therefore does not require a cursor when it is executed. *Compare with* cursor function.

nonvariant function. A user-defined function that always returns the same value when passed the same arguments. A nonvariant function must not contain SQL statements. *Compare with* variant function.

OAT. *See* "OpenAdmin Tool."

object-relational database. A database that adds object-oriented features to a *relational database*, including support for *user-defined data types*, *user-defined routines*, *user-defined casts*, *user-defined access methods*, and *inheritance*.

OID. Object identifier. A numerical value that identifies an MIB, an MIB table, a parameter (column) in an MIB table, or an object (row) in an MIB table.

OLTP. Online transaction processing. *See also* online transaction processing application.

online transaction processing application (OLTP). Characterized by quick, indexed access to a small number of data items. The applications are typically multiuser, and response times are measured in fractions of seconds. *See also* decision-support application.

opaque data type. A *data type* whose inner structure is not visible to the database server. Opaque types that are not built-in need *user-defined routines* and *user-defined operators* that work on them. Synonym for *base type* and *user-defined base type*.

OpenAdmin Tool (OAT). A PHP-based Web-browser administration tool that provides the ability to administer multiple database server instances from a single location.

operational table. A logging permanent table that uses light appends for fast update operations. Operational tables do not perform record-by-record logging.

operator. In an SQL statement, a keyword (such as UNITS or UNION) or a symbol (such as =, >, <, +, -, or *) that invokes an *operator function*. The operands to the operator are arguments to the *operator function*.

operator class. An association of *operator-class functions* with a *secondary access method*. The database server uses an operator class to optimize queries and build an index of that secondary access method.

operator-class function. One of the operator-class support functions or *operator-class strategy functions* that constitute an *operator class*. For user-defined operator classes, the operator-class functions are *user-defined functions*.

operator function. An arithmetic function that has a corresponding operator symbol. An operator function processes one to three arguments and returns a value. For example, the **plus()** function corresponds to the "+" operator symbol.

operator-class strategy function. An *operator-class function* that can appear as a *filter* in a query. The query optimizer uses the strategy functions to determine if an *index* of a particular *secondary access method* can be used to process the filter. You register operator-class strategy functions in the STRATEGIES clause of the CREATE OPCLASS statement.

operator-class support function. An *operator-class function* that a *secondary access method* uses to build or search an *index*. You register operator-class support functions in the SUPPORT clause of the CREATE OPCLASS statement.

owner-privileged. A class of SPL routines that any user can create who has Resource database privileges.

packed decimal. A storage format that represents either two decimal digits or a sign and one decimal digit in each byte.

page. The physical unit of disk storage and basic unit of memory storage that the database server uses to read from and write to Informix databases. Page size is fixed for a particular operating system and platform. A page is always entirely contained within a chunk.

parallel database query (PDQ). The execution of SQL queries in parallel rather than sequential order. The tasks that a query requires are distributed across several processors. This type of distribution enhances database performance.

parallelizable routine. A routine that can be executed within a parallel database query statement.

parallel-processing platform. A parallel-processing platform is a set of independent computers that operate in parallel and communicate over a high-speed network, bus, or interconnect. *See also* symmetric multiprocessing system.

parameter. The specification of a variable that can be changed, passed, or returned. A parameter might include a name, type, or direction. Parameters are used for operations, messages, and events. *See also* configuration file, input parameter, and routine signature.

parent table. The referenced table in a referential constraint. *See also* child table.

participant. In Enterprise Replication, the data (database, table, and columns) to replicate and the database servers to which the data replicates.

PDQ. Parallel database query.

PDQ priority. Determines the amount of resources that a database server allocates to process a query in parallel. These resources include memory, threads (such as scan threads), and sort space. The level of parallelism is established by using the PDQPRIORITY environment variable or various database server configuration parameters (including PDQPRIORITY and MAX_PDQPRIORITY) or dynamically through the SET PDQPRIORITY statement.

phantom row. A row of a table that is initially modified or inserted during a transaction but is subsequently rolled back. Another user can see a phantom row if the isolation level is Informix Dirty Read or ANSI compliant Read Uncommitted. No other isolation level allows a user to see a changed but uncommitted row.

physical log. A set of contiguous disk pages in shared memory where the database server stores an unmodified copy (before-image) of pages before the changed pages are recorded. The pages in the physical log can be any database server page except a blob space blobpage.

pointer. A data element or variable that holds the address of a data object or a function.

precision. An attribute of a number that describes the total number of binary or decimal digits, excluding the sign. The sign is considered positive if the value of a number is zero. *See also* scale.

predefined opaque data type. An opaque data type for which the database server provides the data type definition. *See also* LVARCHAR and pointer.

predicate. *See* filter.

predicate lock. A lock held on index keys that qualifies for a predicate. In a predicate lock, exclusive predicates consist of a single key value, and shared predicates consist of a query rectangle and a scan operation such as inclusion or overlap.

prepared statement. An SQL statement that is generated by the PREPARE statement from a character string or from a variable that contains a character string. This feature allows you to form your request while the program is executing without having to modify and recompile the program.

prepare script. A file containing SQL statements that describe the DataBlade module. There are two types of prepare scripts:

- The script called **prepare.sql** contains information about the module that is not language-specific.
- Scripts with names in the format **prepare.locale.sql** contain language-specific information such as the module and vendor descriptions.

preprocessor. A routine that processes source code before the code is compiled, resulting in altered source code.

primary access method. An access method whose *routines* access a *table* with such operations as inserting, deleting, updating, and scanning. *See also* secondary access method.

primary key. In a relational database, a key that uniquely identifies one row of a database table. A table cannot be defined as a parent unless it has a unique key or primary key.

primary-key constraint. Specifies that each entry in a column or set of columns contains a unique non-null value.

primary server. A database server participating in a high-availability configuration. A primary server permits read and write access from client applications and owns the logical logs that are sent to secondary servers.

privilege. The right to use or change the contents of a database, table, table fragment, or column.

procedure. A *routine* that does not return values. *See also* user-defined procedure.

procedure overloading. *See* routine overloading.

process. An instance of a program running on a system and the resources that it uses.

projection. Taking a subset of the columns in a table. Projection is implemented through the Projection clause of the SELECT statement and returns some of the columns and all of the qualifying rows of a table. The set of columns in the Projection clause is sometimes called the select list.

promotable lock . A lock that can be changed from a shared lock to an exclusive lock. *See also* update lock.

protocol. A set of rules controlling the communication and transfer of data between two or more devices or systems in a communication network.

purpose function. One of a set of functions that an access method uses to create, search, and drop indexes, and to insert entries into an index, delete from an index, and so on.

query. A request to the database to retrieve data that meet certain criteria, usually made with the SELECT statement.

query optimization information statements. The SQL statements that are used to optimize the performance of queries. These statements include SET EXPLAIN, SET OPTIMIZATION, and UPDATE STATISTICS.

query optimizer. A server facility that estimates the most efficient plan for executing a query in the database server. The optimizer considers the CPU cost and the I/O cost of executing a plan.

range fragmentation. A distribution scheme that distributes data in table fragments that contain a specified key range. This technique can eliminate scans of table fragments that do not contain the required rows, making queries faster.

range rule. A user-defined algorithm for *expression-based fragmentation*. It defines the boundaries of each fragment in a table using SQL relational and logical operators. Expressions in a range rule can use the following restricted set of operators: >, <, >=, <=, and the logical operator AND.

raw device. *See* unbuffered disk I/O.

raw disk. *See* unbuffered disk I/O.

raw table. A nonlogged permanent table that uses *light appends*.

read committed. A level of isolation that the SET TRANSACTION statement can specify, in which a user can view rows that are currently committed at the moment of the query request, but cannot view rows that were changed as part of a currently uncommitted transaction. This is the default isolation level for databases that are not ANSI compliant. *See also* committed read.

read uncommitted. An ANSI-compliant level of isolation, set with the SET TRANSACTION statement, that does not account for locks. This allows a user to view any existing rows, even rows that can be altered within currently uncommitted transactions. Read uncommitted is the lowest level of isolation (no isolation at all), and is thus the most efficient. *See also* dirty read.

record. *See* row.

Record-ID (RID). A four-byte RSAM entity that describes the logical position of the record within a fragment. Not the same as rowid.

recovery point objective (RPO). In disaster recovery planning, the age of the data you want to restore in the event of a disaster.

recovery time objective (RTO). In disaster recovery planning, the length of time you can afford to be without your systems.

referential constraint. The relationship between columns within a table or between tables. The relationship is established with columns known as *foreign keys*.

registering. In a database, the process of storing information about user-defined *database objects* in the *system catalog tables* of a database. For example, the CREATE FUNCTION and CREATE PROCEDURE statements register a *user-defined routine* in a database. Also, BladeManager registers groups of user-defined objects that are packaged as DataBlade modules.

relational database. A database that uses table structures to store data. Data in a relational database is divided across tables in such a way that additions and modifications to the data can be made easily without loss of information.

relational database server. A database server that manages data values that are stored in rows and columns.

remainder page. A page that accommodates subsequent bytes of a long data row. If the trailing portion of a data row is less than a full page, it is stored on a remainder page. After the database server creates a remainder page for a long row, it can use the remaining space in the page to store other rows. Each full page that follows the home page is referred to as a big-remainder page.

remote connection. A communications link between the local system and a system or device located on another network, or at a distant site.

remote routine. A routine in a databases of a remote server. *See* subordinate server.

remote server. *See* subordinate server.

remote table. In a distributed query, a table in a database of a server that is not the local database server. *See also* coordinating server and subordinate server.

repeatable read. An Informix and ANSI level of isolation available with the Informix SET ISOLATION statement or the ANSI compliant SET TRANSACTION statement, which ensures that all data values read during a transaction are not modified during the entire transaction. Transactions under ANSI repeatable read are also known as serializable. Informix repeatable read is the default level of isolation for ANSI compliant databases. *See also* serializable.

replicate. In Enterprise Replication, pertains to participants and various attributes of how to replicate the data, such as frequency and how to handle any conflicts during replication. *See also* "master replicate" on page 1-15 and "shadow replicate" on page 1-22.

replication server. An Informix database server that participates in Enterprise Replication.

reserved pages. The first 12 pages of the initial chunk of the root dbspace. Each reserved page stores specific control and tracking information that the database server uses.

reserved word. A word that has been set aside for special use in the SQL standard.

response file. An ASCII file that can be customized with the setup and configuration data that automates an installation. The setup and configuration data would have to be entered during an interactive install, but with a response file, the installation can proceed without any user intervention. *Contrast with “manifest file” on page 1-15.*

R-tree index. A type of index that uses a tree structure based on overlapping bounding rectangles to speed access to spatial and multidimensional data types. *See also* bitmap index and B-tree index.

role. An authorization identifier to which access privileges on objects in the database can be granted. A user can be granted one or more roles, and can hold the privileges of the current role, in addition to any PUBLIC or individually-granted privileges that the user holds.

role separation. A database server installation option that allows different users to perform different administrative tasks.

roll back. To remove changes that were made to database files under commitment control since the last commitment boundary. *See also* transaction and commit work.

root dbspace. The initial *dbspace* that the database server creates. The root dbspace contains reserved pages and internal tables that describe and track all other dbspaces, blobspaces, sbspaces, tblspaces, chunks, and databases.

root node. A single index page that contains node pointers to *branch nodes*. The database server allocates the root node when you create an index for an empty table.

root page. The topmost level in a tree structure. In an R-tree index, the root page can have zero or more branch pages or leaf pages below it, depending on the size of the R-tree index.

root supertype. The *named row data type* at the top of a *type hierarchy*. A root supertype has no *supertype* above it.

round-robin fragmentation. A method of partitioning a table or index into fragments in which the database server balances the number of rows in each fragment. As more rows are inserted, the database server determines the fragment in which they should reside.

routine. A group of program statements that perform a particular task. A routine can be a *function* or a *procedure*. All routines can accept arguments. *See also* user-defined routine (UDR).

routine modifier. A keyword in the WITH clause of a CREATE FUNCTION, CREATE PROCEDURE, ALTER FUNCTION, ALTER PROCEDURE, or ALTER ROUTINE statement that specifies a particular attribute or usage of a *user-defined routine*.

routine overloading. The ability to assign one name to multiple *user-defined routines* and specify *parameters* of different data types on which each routine can operate. An overloaded routine is uniquely defined by its *routine signature*.

routine resolution. The process that the database server uses to determine which *user-defined routine* to execute, based on the *routine signature*. *See also* routine overloading.

routine signature. The information that the database server uses to uniquely identify a user-defined routine. *See also* routine overloading.

row. A group of related items of information about a single entity across all columns in a database table. In a table of customer information, for example, a row contains information about a single customer. A row is sometimes referred to as a *record* or *tuple*. In an object-relational model, each row of a table stands for one *instance* of the subject of the table, which is one particular example of that entity.

row constructor. The type constructor used to construct row data types.

row data type. A complex data type that contains one or more related data fields, of any data type except IDSECLABEL, that form a template for a record. The data in a row type can be stored in a row or column. *See also* named row data type and unnamed row data type.

row variable. An IBM Informix ESQ/C *host variable* or *SPL variable* that holds an entire *row type* and provides access to the individual *fields* of the row.

rowid. In nonfragmented tables, the database server assigns a unique rowid that defines the physical location of a row.

RS secondary server. A remote standalone secondary server participating in a high-availability configuration. RS secondary servers can be geographically distant from the primary server, serving as remote back-up servers in disaster-recovery scenarios. Each RS secondary server maintains a complete copy of the database, with updates transmitted asynchronously from the primary server over secure network connections. *See also* “SD secondary server.”

runtime environment. The hardware and operating-system services available at the time a program runs.

runtime error. An error that occurs during program execution. *Compare with* compile-time error.

sbspace. A logical storage area that contains one or more chunks that store only BLOB and CLOB data.

scale. The number of digits to the right of the decimal place in DECIMAL notation. The number 14.2350 has a scale of 4 (four digits to the right of the decimal point). *See also* precision.

schema. A collection of database objects such as tables, views, indexes, or triggers that define a database. A database schema provides a logical classification of database objects.

scope of reference. The portion of a *routine* or application program where an *identifier* can be accessed. Three possible scopes exist: local (applies only in a single statement block), modular (applies throughout a single module), and global (applies throughout the entire program). *See also* local variable and global variable.

scroll cursor. A cursor created with the SCROLL keyword that allows you to fetch rows of the active set in any sequence.

secondary access method. (1) An access method whose *routines* access an *index* with such operations as inserting, deleting, updating, and scanning. (2) A set of server functions that build, access, and manipulate an index structure: for example, a B-tree, an R-tree, or an index structure that a DataBlade module provides. Typically, a secondary access method speeds up the retrieval of data.

SD secondary server. A database server participating in a high-availability configuration. The secondary server receives database updates from a primary server but does not permit updates from client applications. *See also* “RS secondary server” on page 1-20.

secure auditing facility. A facility of Informix database servers that lets a database system security officer monitor unusual or potentially harmful user activity. Use the **onaudit** utility to enable auditing of events and to create audit masks. Use the **onshowaudit** utility to extract audit event data for analysis.

Secure Sockets Layer (SSL). A security protocol that provides communication privacy. With SSL, client/server applications can communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

security label. In label-based access control, a database object applied to users and table rows and columns to protect data from unauthorized access. The security label of a user and a table row or column must match for the user to access the data of that particular row or column.

security label component. In label-based access control, a database object that represents one of the criteria that an organization uses to decide who has access to specific data. There are 3 types of components: sets, arrays, and trees. *See also* security label, security policy.

security policy. In label-based access control, a database object that is associated with one or more tables and that defines how LBAC can be used to protect those tables. The security policy defines what security labels can be used, how the security labels are compared to each other, and whether optional behaviors are used. The security policy is composed of one or more security label components. *See also* label-based access control (LBAC), security label, and security label component.

select cursor. A cursor that is associated with a SELECT statement, which lets you scan multiple rows of data, moving data row by row into a set of receiving variables.

selection. A horizontal subset of the rows of a single table that satisfies a specified condition. Selection is implemented through the WHERE clause of a SELECT statement and returns some of the rows and all of the columns in a table. *See also* projection.

selective index. A type of generalized-key index that contains keys for only a subset of a table.

selectivity. The probability that any table row will satisfy a predicate.

selectivity function. A function that calculates the percentage of rows that will be returned by a filter function in the WHERE clause of a query. The optimizer uses selectivity information to determine the fastest way to execute an SQL query.

sequence. A database object that is independent of any one table that automatically generates unique key values based on initial user specifications.

sequential cursor. A cursor that can fetch only the next row in sequence. A sequential cursor can read through a table only once each time the sequential cursor is opened.

serializable. An ANSI-compliant level of isolation set with the SET TRANSACTION statement, ensuring all data read during a transaction is not modified during the entire transaction. *See also* repeatable read.

server locale. The locale that a database server uses when it performs its own read and write operations. The `SERVER_LOCALE` environment variable can specify a nondefault locale. *See also* client locale and locale.

server name. The unique name of a database server, assigned by the database server administrator, that an application uses to select a database server.

server number. A unique number between 0 and 255, inclusive, that a database server administrator assigns when a database server is initialized.

server multiplexer group (SMX) connection. A multiplexed network connection that provides reliable, secure, high-performance communication between servers in a high-availability environment.

server-processing locale. The locale that a database server determines dynamically for a given connection between a client application and a database. *See also* locale.

shadow replicate. A copy of an existing (primary) Enterprise Replication replicate. Shadow replicates allow Enterprise Replication to manage alter and repair operations on replicated tables. *See also* "master replicate" on page 1-15 and "replicate" on page 1-19.

shared library. A library device that is shared among multiple storage management servers. *See also* dynamic link library (DLL).

shared lock. A lock that more than one thread can acquire on the same object. Shared locks allow for greater concurrency with multiple users; if two users have shared locks on a row, a third user cannot change the contents of that row until both users (not just the first) release the lock.

shared memory. A portion of main memory that is accessible to multiple processes. Shared memory allows multiple processes to communicate and access a common data space in memory. Common data does not have to be reread from disk for each process, reducing disk I/O and improving performance. Also used as an Inter-Process Communication (IPC) mechanism to communicate between two processes running on the same computer.

shared-object file. A *library* that is not linked to an application at compile time but instead is loaded into memory by the operating system as needed. Several applications can share access to the loaded shared-object file. *See also* dynamic link library (DLL) and shared library.

signature. *See* routine signature.

silent installation. An installation option that allows users to install the product without having to manually specify the installation information during the installation process. The users provide an INI file that contains the installation information that is used automatically by the installation application.

simple large object. A *large object* that is stored in a *blob*space or *db*space is not recoverable and does not obey transaction isolation modes. Simple large objects include TEXT and BYTE data types.

simple predicate. A search condition in the WHERE clause that has one of the following forms: $f(\text{column}, \text{constant})$, $f(\text{constant}, \text{column})$, or $f(\text{column})$, where f is a binary or unary function that returns a Boolean value (true, false, or unknown).

single-byte character. A character that uses one byte of storage. Because a single byte can store values in the range of 0 to 255, it can uniquely identify 256 characters. With these code sets, an application can assume that one character is always stored in one byte. *See also* 8-bit character and multibyte character.

singleton implicit transaction. A single-statement transaction that does not require either a BEGIN WORK or a COMMIT WORK statement. This type of transaction can occur only in a database that is not ANSI compliant, but that supports transaction logging. *See also* explicit transaction and implicit transaction.

singleton SELECT. A SELECT statement that returns a single row.

SLV. statement local variable.

smart large object. A *large object* that is stored in an *sbspace*, which has read, write, and seek properties similar to a UNIX file, is recoverable, obeys transaction isolation modes, and can be retrieved in segments by an application. Smart large objects include BLOB and CLOB data types.

SMI. System monitoring interface.

SMP. Symmetric multiprocessing system.

SNMP. Simple Network Management Protocol. A communication protocol that you use to manage components on a network.

SSL. *See* Secure Sockets Layer.

source file. A file of programming code that is not compiled into machine language. *See also* compile.

source type. The data type from which a DISTINCT type is derived.

SPL. *See* Stored Procedure Language.

SPL function. An *SPL routine* that returns one or more values.

SPL procedure. An *SPL routine* that does not return a value.

SPL routine. A *user-defined routine* that is written in Stored Procedure Language (*SPL*). Its name, parameters, executable format, and other information are stored in the system catalog tables of a database. An *SPL routine* can be an *SPL procedure* or an *SPL function*.

SPL variable. A *variable* that is declared with the DEFINE statement in an *SPL routine*.

SQL. Acronym for *Structured Query Language*. SQL is a database query language that was developed by IBM and standardized by ANSI. Informix relational database management products are based on an implementation of ANSI-standard SQL.

SQL API. An *application programming interface* that allows you to embed SQL statements directly in an application. The embedded-language product IBM Informix ESQL/C is an example of an SQL API. *See also* host variable.

SQLCA. SQL Communications Area. SQLCA is a data structure that stores information about the most recently executed SQL statement. The result code returned by the database server to the SQLCA is used for error handling by Informix SQL APIs.

sqllda. Acronym for *SQL descriptor area*. A dynamic SQL management structure that can be used with the DESCRIBE statement to store information about database columns or host variables used in dynamic SQL statements. The **sqllda** structure is an Informix-specific structure for handling dynamic columns. It is available only within an IBM Informix ESQL/C program. *See also* descriptor and system-descriptor area.

sqlhosts. A file that identifies the types of connections that the database server supports.

SQLSTATE. A variable that contains status values about the outcome of SQL statements.

stack operator. Operator that allows programs to manipulate values that are on the stack.

statement. An instruction in a program or procedure.

statement block. A unit of SPL program code that performs a particular task and is usually marked by the keywords `begin` and `end`. The statement block of an *SPL routine* is the smallest scope of reference for program variables.

statement identifier. An embedded variable name or SQL statement identifier that represents a data structure defined in a `PREPARE` statement. It is used for dynamic SQL statement management by Informix SQL APIs.

statement label. In SPL routines, an SQL identifier that enables the `GOTO` label statement to transfer program control to the first executable statement that follows the declaration of the specified statement label.

statement local variable (SLV). A SPL, C, or Java function that is invoked in the `WHERE` clause of a query, which can declare one or more statement-local variables that are visible in other parts of the same query, including its subqueries. You can assign to the SLV the values of `OUT` or `INOUT` parameters of the function.

static table. A nonlogging, read-only permanent table.

status variable. A program variable that indicates the status of some aspect of program execution. Status variables often store error numbers or act as flags to indicate that an error has occurred.

storage space. A *dbspace*, *blobspace*, or *sbspace* that is used to hold data.

stored procedure. See *SPL routine*.

Stored Procedure Language (SPL). An Informix extension to SQL that provides flow-control features such as sequencing, branching, and looping. See also *SPL routine*.

strategy function. See *operator-class strategy function*.

string. A sequence of text characters. The details of string representation depend on implementation, and might include character sets that support international characters and graphics.

subordinate server. Any database server in a distributed query that did not initiate the query. See also *coordinating server*.

subquery. In SQL, a subselect used within a predicate. For example, a select-statement within the `WHERE` or `HAVING` clause of another SQL statement.

subscript. An integer or variable whose value selects a particular element in a table or array.

substring. A part of a character string.

subtable. A *typed table* that inherits properties (column definitions, constraints, triggers) from a *supertable* above it in the *table hierarchy*.

subtype. A *named row data type* that inherits all representation (*data fields*) and behavior (*routines*) from a *supertype* above it in the *type hierarchy* and can add additional fields and routines. The number of fields in a subtype is always greater than or equal to the number of fields in its supertype.

supertable. A *typed table* whose properties (constraints, storage options, triggers) are inherited by a *subtable* beneath it in the *table hierarchy*. The scope of a query on a supertable is the supertable and its subtables.

supertype. A *named row data type* whose representation (*data fields*) and behavior (*routines*) is inherited by a *subtype* below it in the *type hierarchy*.

support function. See *aggregate support function* and *operator-class support function*.

support routine. See *support function*.

symmetric multiprocessing system (SMP). A system composed of multiple computers that are connected to a single high-speed communication subsystem.

synonym. A name that is assigned to a table, view, or sequence and that can be used in place of the original name. A synonym does not replace the original name; instead, it acts as an alias for the table, view, or sequence.

sysadmin database. A server-level database that supports all the logged databases of a Dynamic Server instance. Its tables store information that the DBSA can use in various administrative operations, including tuning and configuring the database server, monitoring and analyzing resource usage, scheduling recurring maintenance tasks, and logging calls to the SQL Administration API functions.

system call. A call by a program to an operating system subroutine.

system catalog. A group of database tables that contains information about the database itself, such as the names of tables or columns in the database, the number of rows in a table, and the information about indexes and database privileges. *See also* data dictionary.

system-defined cast. A pre-defined *cast* that is known to the database server. Each built-in cast performs automatic conversion between two different *built-in data types*.

system-descriptor area. A dynamic SQL management structure that is used with the ALLOCATE DESCRIPTOR, DEALLOCATE DESCRIPTOR, DESCRIBE, GET DESCRIPTOR, and SET DESCRIPTOR statements to store information about database columns or host variables used in dynamic SQL statements. The structure contains an item descriptor for each column; each item descriptor provides information such as the name, data type, length, scale, and precision of the column. The system-descriptor area is the X/Open standard for handling dynamic columns. *See also* descriptor and SQLDA.

system index. An index that the database server creates to implement a *unique constraint* or a *referential constraint*. A system index is distinct from a *user index*, which a user creates explicitly.

system-monitoring interface (SMI). A collection of tables in the **sysmaster** database that maintains dynamically updated information about the operation of the database server. The tables are constructed in memory but are not recorded on disk. Users can query the SMI tables with the SELECT statement of SQL.

table fragment. Zero or more rows that are grouped together and stored in a dbspace that you specify when you create the fragment. A virtual table fragment might reside in an *sbspace* or an *extspace*.

table fragmentation. A method of separating a table into potentially balanced fragments to distribute the workload and optimize the efficiency of the database operations. Also known as data partitioning. Table-fragmentation methods (also known as distribution schemes) include *expression-based*, *hybrid*, *range*, *round-robin*, and *system-defined hash*.

table hierarchy. A relationship you can define among *typed tables* in which *subtables* inherit the behavior (constraints, triggers, storage options) from *supertables*. Subtables can have additional constraint definitions, storage options, and triggers.

table inheritance. The property that allows a *typed table* to inherit the behavior (constraints, storage options, triggers) from a *typed table* above it in the *table hierarchy*.

target table. The underlying base table that a violations table and diagnostics table are associated with. You use the START VIOLATIONS TABLE statement to create the association between the target table and the violations and diagnostics tables.

tblspace. A table space, which is a logical collection of *extents* that are assigned to a table. A table space contains all the disk space that is allocated to a given table or table fragment and includes pages allocated to data and to indexes, pages that store TEXT or BYTE data in the dbspace, and bitmap pages that track page use within the extents.

template. A mechanism to set up and deploy Enterprise Replication for a group of tables on one or more servers.

temporary dbspace. A dbspace used to store temporary tables or other data that need not be saved between sessions.

terabyte. For processor storage, real and virtual storage, and channel volume, 2 to the 40th power or 1 099 511 627 776 bytes. For disk storage capacity and communications volume, 1 000 000 000 000 bytes.

text data type. A *data type* for a *simple large object* that stores text and can be as large as 2³¹ bytes.

thread. A stream of computer instructions that is in control of a process. In some operating systems, a thread is the smallest unit of operation in a process. Several threads can run concurrently, performing different jobs. *See also* multithreading and user thread.

trace. To keep a running list of the values of program variables, arguments, expressions, and so on, in a program or SPL routine.

transaction. A collection of one or more SQL statements that is treated as a single unit of work. If one statement in a transaction fails, the entire transaction can be rolled back (canceled). If the transaction is successful, the work is committed and all changes to the database from the transaction are accepted. *See also* explicit transaction, implicit transaction, and singleton implicit transaction.

transaction logging. The process of keeping records of transactions. *See also* logical log.

transaction mode. The method by which constraints are checked during transactions. You use the SET statement to specify whether constraints are checked at the end of each data manipulation statement or after the transaction is committed.

Transient type. An unnamed complex data type that is used to hold query results or is part of a temporary table.

trigger. A database object that is associated with a single table or with one or more base tables of a view and that associates a triggering event with a triggered action. The triggered action consists of a set of SQL statements or function calls that is issued when the triggering event (a DELETE, INSERT, UPDATE, or SELECT operation) specifies the associated table or view on which the trigger is defined.

tuple. *See* row.

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* heterogeneous commit.

type constructor. An SQL keyword that indicates to the database server the type of complex data to create (for example, LIST, MULTISSET, ROW, SET).

typed collection variable. An ESQL/C *collection variable* or *SPL variable* that has a defined *collection data type* associated with it and can only hold a *collection* of its defined type. *See also* untyped collection variable.

typed table. A table that is constructed from a *named row data type* and whose rows contain instances of that *row data type*. A typed table can be used as part of a *table hierarchy*. The columns of a typed table correspond to the *fields* of the named row type.

type hierarchy. A relationship that you define among *named row data types* in which *subtypes* inherit representation (data *fields*) and behavior (*routines*) from *supertypes* and can add more fields and routines.

type inheritance. The property that allows a *named row data type* or *typed table* to inherit representation (data *fields*, columns) and behavior (*routines*, operators, rules) from a named row type above it in the *type hierarchy*.

type substitutability. The ability to use an instance of a subtype when an instance of its supertype is expected.

typical installation. An installation option in which the products are installed with all features and default installation settings. *See also* "custom installation" on page 1-6.

UDA. User-defined aggregate

UDF. User-defined function

UDR. User-defined routine

UDT. User-defined data type

unattended installation. An installation that does not require any user interaction. *See* "silent installation" on page 1-22.

unbuffered disk I/O. Disk I/O that is controlled directly by the database server instead of the operating system. This direct control helps improve performance and reliability for updates to data. Unbuffered I/O is supported by character-special files on UNIX and by both unbuffered files and the raw disk interface on Windows.

Uncommitted Read. *See* Read Uncommitted.

unique constraint. Specifies that each entry in a column or set of columns has a unique value.

unique index. An index that prevents duplicate values in the indexed column.

unique key. *See* primary key.

unlock. To release an object or system resource that was previously locked and return it to general availability.

unnamed row data type. A *row data type* created with the ROW constructor that has no defined name and no inheritance properties. Two unnamed row data types are equivalent if they have the same number of *fields* and if corresponding fields have the same *data type*, even if the fields have different names.

untyped collection variable. A generic ESQL/C *collection variable* or *SPL variable* that can hold a *collection* of any *collection data type* and takes on the data type of the last collection assigned to it. *See also* typed collection variable.

updatable view. A view whose underlying table can be modified by inserting values into the view.

update. The process of changing the contents of one or more columns in one or more existing rows of a table.

update lock. A promotable lock that is acquired during a SELECT...FOR UPDATE. An update lock behaves like a shared lock until the update actually occurs, and it then becomes an exclusive lock. It differs from a shared lock in that only one update lock can be acquired on an object at a time.

user-defined aggregate (UDA). An aggregate function that is not provided by the database server (is not built in) that includes extensions to built-in aggregates and newly defined aggregates. The database server manages all aggregates.

user-defined cast. A *cast* that a user creates with the CREATE CAST statement. A user-defined cast typically requires a *cast function*. A user-defined cast can be an *explicit cast* or an *implicit cast*.

user-defined data type (UDT). A *data type* that you define for use in a *relational database*. You can define *opaque data types* and *distinct data types*.

user-defined function (UDF). A *user-defined routine* that returns at least one value. You can write a user-defined function in SPL (*SPL function*) or in an external language that the database server supports (*external function*).

user-defined procedure. A *user-defined routine* that does not return a value. You can write a user-defined procedure in SPL (*SPL procedure*) or in an external language that the database server supports (*external procedure*).

user-defined routine (UDR). A routine that users write and register in the system catalog tables of a database, and that an SQL statement or another routine can invoke. A user-defined routine is written in SPL or in an external language (external routine) that the database server supports.

user-defined statistics. Information about the opaque data type values in your database that is collected by the UPDATE STATISTICS statement, which calls user-defined functions to calculate the statistics. The optimizer uses these statistics to determine the fastest way to execute an SQL query.

user index. An index that a user creates explicitly with the CREATE INDEX statement. *Compare with* system index.

user thread. User threads include session threads (called **sqlexec** threads) that are the primary threads that the database server runs to service client applications. User threads also include a thread to service requests from the **onmode** utility, threads for recovery, and page-cleaner threads. *See* thread.

value object. A self-contained binary object that provides standard interfaces to its callers. Value objects can be used in client applications.

variable. A representation of a changeable value. *Compare with* pointer.

variant function. A *user-defined function* that might return different values when passed the same arguments. A variant function can contain SQL statements. *Compare with* nonvariant function.

variant routine. A routine that can return different values when it is invoked with the same arguments.

view. A dynamically controlled subset of the columns of one or more database tables. A view can give the programmer control over what information the user sees and manipulates and represents a virtual table that holds the results of a specified SELECT statement.

view folding. A query plan strategy for joined queries in which one or more of the data sources is a view, by which the query optimizer incorporates the view definitions into the main query. For queries in which this technique is possible, view folding can significantly improve performance, as compared to materializing the view as a temporary table.

violations table. A special table that holds rows that fail to satisfy constraints and unique index requirements during data manipulation operations on base tables. You use the START VIOLATIONS TABLE statement to create a violations table and associate it with a base table.

virtual column. A derived column of information, created with an SQL statement that is not stored in the database. For example, you can create virtual columns in a SELECT statement by arithmetically manipulating a single column, such as multiplying existing values by a constant, or by combining multiple columns, such as adding the values from two columns.

virtual-column index. A type of *generalized-key index* that contains keys that are the result of an expression.

virtual processor. A multithreaded process that makes up the database server and is similar to the hardware processors in the computer. It can serve multiple clients and, where necessary, run multiple threads to work in parallel for a single query.

virtual table. A table created to access data in an external file, external DBMS, smart large object, or in the result set of an iterator function in a query. The database server does not manage external data or directly manipulate data within a smart large object. The Virtual-Table Interface allows users to access the external data in a virtual table using SQL DML statements and join the external data with Dynamic Server table data.

white space. A series of one or more space characters. The GLS locale defines the characters that are considered to be space characters. For example, both the TAB and blank might be defined as space characters in one locale, but certain combinations of the CTRL key and another character might be defined as space characters in a different locale.

wide character. A character whose range of values can represent distinct codes for all members of the largest extended character set specified among the supporting locales.

X/Open. An independent consortium that produces and develops specifications and standards for open-systems products and technology, such as dynamic SQL.

X/Open Backup Services Application Programming Interface (XBSA). Provides a programming interface and functions that manage backup and restore operations. XBSA can connect storage managers with the database server.

zoned decimal. A data representation that uses the low-order four bits of each byte to designate a decimal digit (0 through 9) and the high-order four bits to designate the sign of the digit.

Appendix. Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

Accessibility features for IBM Informix Dynamic Server

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility Features

The following list includes the major accessibility features in IBM Informix Dynamic Server. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.
- The attachment of alternative input and output devices.

Tip: The IBM Informix Dynamic Server Information Center and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard Navigation

This product uses standard Microsoft Windows navigation keys.

Related Accessibility Information

IBM is committed to making our documentation accessible to persons with disabilities. Our publications are available in HTML format so that they can be accessed with assistive technology such as screen reader software. The syntax diagrams in our publications are available in dotted decimal format.

You can view the publications for IBM Informix Dynamic Server in Adobe Portable Document Format (PDF) using the Adobe Acrobat Reader.

IBM and Accessibility

See the *IBM Accessibility Center* at <http://www.ibm.com/able> for more information about the commitment that IBM has to accessibility.

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 or send inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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 which has been exchanged, should contact:

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

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

The licensed program described in this information 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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains 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 contains 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. 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 IBM's application programming interfaces.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos 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.



Printed in USA

GC23-9431-00

