

# Glossary

If you do not find the term you are looking for here, try <http://whatis.techtarget.com/>, <http://www.webopedia.com/> or <http://www.hiddenlab.com/acronym/>.

**ACID:** Acronym for four properties guaranteed by *transactions*:

**Atomicity:** Either the entire transaction executes, or none of it does.

**Consistency:** No transaction can break the *integrity constraints* of the database.

**Isolation:** no operation outside the transaction can see its intermediate results.

**Durability:** Once the user is notified of success, the operation cannot be rolled back.

In an ACID-compliant system, all data changes follow ACID rules.

**Anti-join:** A join which restricts output to rows of the first table that have no matching rows in the second. Examples: NOT IN(), NOT EXISTS(), *exclusion join*.

**API:** Application Program Interface

**Applet:** Component that executes in a web browser, or other applications or devices that support the applet programming model

**Application client:** First-tier client Java component that executes in its own JVM and has access to some J2EE platform APIs (JNDI, JDBC, RMI-IIOP, JMS)

**Asynchronous replication:** Changes in an updatable master table are copied to replication tables at predetermined times or intervals.

**Attribute:** A named object property and set of possible property values

**Binding:** Association of a name with an object (or pointer to it)

**BSF:** Bean Scripting Framework

**Callback method:** Component method called by a container to notify the component of important events in its life cycle

**Caller principal:** Caller or principal that identifies the invoker of enterprise bean method

**Candidate key:** A key which could be a primary key because all values are unique and it has no redundant information. It is a *superkey* with no redundant information in it.

**CGI:** Common Gateway Interface

**Commit:** Make database changes permanent

**Concurrent insert:** An insert that executes while other threads are reading from the insert table

**Container:** Entity that provides life cycle management, security, deployment, runtime and other specific services to components

**Context:** Set of name-to-object bindings, e.g. in a Java environment

**CORBA:** Common Object Request Broker Architecture

**Correlated subquery:** a subquery which refers to a column referenced in the outer query

**COS:** *CORBA* Common Object Services name service

**Covering index:** Index on all columns referenced in SELECT, JOIN, and WHERE clauses of a query

**CSS:** Cascading Style Sheet, used with *HTML* and *XML* documents to add defined styles for rendering mechanisms like browsers.

**CSV:** Comma-separated values data format.

**CTS:** Compatibility Test Suite

**Cursor:** Temporary database object which provides row-by-row access to a query result.

**Cyclic dependency:** Exists in a table if for any  $i$  and  $j$ , rows  $(A_i, B_i, C_j)$ ,  $(A_i, B_j, C_i)$ , and  $(A_j, B_i, C_i)$  occur but row  $(A_i, B_i, C_i)$  does not occur; therefore reconstituting the table after decomposing it with projections of degree 2 creates spurious rows.

**Database:** Collection of related table and other data-related objects

**DBMS:** Database Management System

**DCOM:** Distributed Component Object Model, a widely followed standard

**DDL:** Data Description Language: the set of all SQL statements that define schemas, objects within them, and user's rights to them

**Dereference:** Return the value of a variable from a reference or pointer to the variable

**Determinant:** One or more key columns on which other columns *functionally depend*, for example if the city is Calgary then the province is Alberta.

**Directory-enabled application:** Application that uses a naming or *directory* service

**Directory Object:** Object in a *Directory Service*

**Directory Service:** *Naming system* which manages named objects (eg add, delete, find, update) and their attributes

**Dirty read:** Reading of row changes which have not yet been *committed*

**Distributed database:** A collection of multiple, logically interrelated databases distributed over a computer network

**Distributed database management system:** software system that transparently manages distributed databases

**Distributed processing:** process of an agent or application (for example a client-server app) distributing its tasks among different computers on a network

**Distributed transaction:** *Transaction* that references multiple servers or network nodes

**DML:** Data Manipulation Language: the set of all *SQL* statements that store, alter or retrieve database data

**DOM:** Domain Object Model, a tree of objects with interfaces for traversing the tree and writing an *XML* version of it as defined by the *W3C* specification.

**Domain:** Definition of a data type for a database.

**Domain integrity:** Validity of data for its domain, e.g., an age may not be negative.

**DNS:** Internet Domain Name System

**DRBD:** Acronym for Distributed Replication Block Device

**DSN:** Acronym for Data Source Name

**DTD:** Document Type Definition describing structure and properties of a class of *XML* files

**EAR file:** *JAR* archive containing a *J2EE* application

**EJB:** Enterprise JavaBean™

**Entity:** In relational database theory, the logical analogue of a table.

**Entity integrity:** Absence of duplicate rows

**Entity bean:** *EJB* that represents database data

**Equi-join:** A join of relations (tables) on identical column values

**Exclusion join:** LEFT JOIN combined with a WHERE clause which excludes NULL values on the right side of the join.

**Filter:** Object that can transform the header and/or content of a request or response

**FTP:** Acronym for File Transfer Protocol

**Full and partial dependency:** If there is only one value of column C for any set of column A and B values, C is *fully* dependent on A and B; if column D depends only on column A or B, that is a *partial* dependency.

**Functional dependency:** `column1` is functionally dependent on `column2` if for each value of `column2` there is no more than one value of `column1`, in which case `column2` is a *determinant* of `column1`.

**GIS:** Acronym for Geographic Information Systems

**Heterogeneous distributed database system:** A distributed database system where databases are of different types

**Homogeneous distributed database system:** A distributed database system where all databases are of the same type

**HTML:** Hypertext Markup Language

**HTTP:** HyperText Transfer Protocol

**HTTPS:** *HTTP* layered over the SSL protocol

**IDL:** Interface Definition Language

**IIOP:** *CORBA* Internet Inter-ORB Protocol

**Information schema:** Set of views that describe *database* objects

**INNER JOIN:** A relational database operation which selects rows from two tables such that the value in one column of the first table also appears in a specified column of the second table.

**Inode:** Unix data structure for a file, with properties describing file name(s), user and group ownership, read/write/execute permissions, and type.

**Integrity constraints:** Rules guaranteeing *entity integrity*, *referential integrity* and *domain integrity*.

**IP:** Internet Protocol

**IPC:** Inter Process Communication

**Isolation level:** Degree to which (i) the intermediate state of the data being modified by a transaction is visible to concurrent transactions and (ii) data being modified by other transactions is visible to it

**J2EE:** Java 2 Enterprise Edition

**J2SDK:** Java 2 Software Development Kit

**JAR file:** Platform-independent Java archive file

**JDBC:** Java Database Connectivity

**JNDI:** Java Naming and Directory Interface

**JNLP:** Java Network Launching Protocol

**JOIN:** Combination of rows FROM two or more relations (tables)

**JRMP:** Acronym for Java Remote Method Protocol

**JSP:** JavaServer Page

**LAMP:** Acronym for Linux, Apache, MySQL, PHP

**LAMPS:** Acronym for LAMP plus SSL

**LDAP:** Lightweight Directory Access Protocol

**LEFT JOIN:** An OUTER JOIN which returns all rows from the table named *before* the JOIN keyword, combined with matching rows or NULLs from the table named *after* the JOIN keyword.

**Location transparency:** A user can refer to any database object in one way, regardless of its location.

**LRU:** Acronym for Least Recently Used

**MD5:** *Message Digest* number 5 value, a 128-bit value similar to a checksum

**Multi-valued dependency:** A relationship amongst three or more columns A, B, C, ... where values of A depend on B, C, ... but values of B, C, ... are mutually independent.

**Mutual independence:** Relationship between two columns where neither is a *determinant* of the other.

**MUTEX:** Mutual exclusion object with two states, locked and unlocked, that allows multiple threads to synchronize access to a shared resource

**MVCC:** Multi-version Concurrency Control updates tables and rows without the overhead of row-level locking.

**Namespace:** Set of names in a naming system

**Naming System:** Connected set of *contexts* of the same type and a set of common operations (eg lookup)

**NATURAL JOIN:** A equi-join of two relations (tables) on columns of the same name and type, showing only one copy of the shared column(s).

**Non-repeatable read:** A read of a row reflects uncommitted differences from a previous read

**ODBC:** Open Database Connectivity

**Object:** An encapsulated set of data and methods

**OLAP:** Acronym for Online Analytical Processing

**OLTP:** Acronym for Online Transaction Processing

**OpenGIS:** Registered Trademark of Open Geospatial Consortium, Inc (OGC), denoting specifications for encoding geospatial information.

**ORB:** Object Request Broker

**ORPC:** Object Remote Procedure Call

**OUTER JOIN:** a join containing all rows from one table and matching rows, or NULL if no matching row exists, from another table. It may be LEFT, RIGHT or FULL. If LEFT, all rows from the table named *before* the JOIN keyword are returned, along with matching rows or NULLs from the table named *after* the JOIN keyword. If RIGHT, vice-versa. If FULL, the result combines LEFT and RIGHT JOINS and eliminates duplicates.

**Outer query:** A query containing a subquery (or inner query)

**Perl:** Practical Extraction and Report Language

**Pessimistic concurrency:** Lock rows at the data source to prevent users from inducing concurrency errors, for example a duplicate primary key value.

**Phantom read:** Read of an inserted row where the changes have not yet been committed

**PHP:** Hypertext processing language

**POA:** Portable Object Adapter, a CORBA standard for building server-side applications that are portable across different ORBs

**POP:** Post Office Protocol

**POSIX:** Portable Operating System Interface, a set of IEEE standards for application portability between Unix variants

**Primary key:** Single attribute (field, column), or collection of them, which uniquely identify a row and which may never be NULL.

**Projection:** A subset of the columns in a table

**Query:** A request for the retrieval of data, using the SELECT command, or a SQL command that inserts, updates or deletes data.

**RAID:** Redundant array of inexpensive disks

**Replication:** the process of copying and maintaining database objects in multiple databases belonging to a *distributed database*

**Ref:** Reference to an SQL structured type value in a table

**Referential integrity:** Consistency between related tables, e.g., all references to data in other tables exist.

**Remote procedure call:** A call to a stored procedure on a remote server

**Rremote transaction:** *Transaction* that references a remote server

**RIGHT JOIN:** OUTER JOIN which returns all rows from the table named *after* the JOIN keyword, combined with matching rows or NULLs from the table named *before* the JOIN keyword.

**RMI:** Remote Method Invocation

**Rollback:** The point in a transaction when all updates to any resources involved in the transaction are reversed.

**Row value constructor:** A sequence of one or more value expressions comparable to a row of a table

**Savepoint:** Point within the current transaction that can be referenced from a rollback

**SAX:** Simple API for XML

**Scalar:** a single quantity, i.e. not an array, vector or table

**Schema:** A collection of objects which model something. A database schema consists of all the objects, and their structures, in a database.

**Semi-join:** A join between two tables which returns one row at most from the first table, and as many matching rows as exist from the second. Examples: IN(), EXISTS().

**SMP:** Symmetric multi-processing computer architecture.

**SMTP:** Simple Mail Transfer Protocol, a widely followed standard

**SNMP:** Simple Network Management Protocol, a widely followed standard

**SOAP:** Simple Object Access Protocol, a widely followed standard

**SPI:** Service Provide Interface

**SQL:** Structrued Query Language

**SQLSTATE:** Five-character ANSI/ODBC code indicating SQL result status. The first two characters denote *class*: 00=success, 01=warning, 02=error, >02=exception.

**SQL/J:** Set of standards that specifies how to embed SQL statements in Java methods, and how to call Java static methods as SQL stored procedures and UDFs

**SSI:** Server-side includes

**SSL:** Secure Socket Layer, security protocol that provides a degree of internet privacy

**Standard deviation:** Square root of the *variance*

**Strong authentication:** A form of computer security where the identities of networked users, clients and servers are verified without transmitting passwords over the network.

**Subquery:** a query contained within another SQL statement

**Superkey:** A combination of columns for which there are no repeating values

**Synchronous replication:** Changes in a table are applied to all replicated copies immediately, as part of the basic transaction.

**Table:** Two-dimensional grid of rows and columns where each row has the same number of cells and is unique, and data in each cell conforms to the column's data domain.

**Table value constructor:** A group of *row value constructors* which all have the same number of cells and in which value types match columnwise.

**Tablespace:** An Oracle-like logical unit for storage segments, i.e. tables and indexes.

**TCP:** Transmission Control Protocol

**Time validity:** A guarantee that every row describes a state of affairs in a valid time period.

**Transaction:** Atomic or logical unit of work that encloses SQL statements which are to be entirely committed or entirely rolled back

**Transaction time validity:** A guarantee that a table is exactly reconstructable as it existed at any instant, or for any period, since its creation.

**Transitive closure** of a binary relation R on a set X is the smallest transitive relation on X that contains R.

**Trigger:** Named database object associated with a table and fired by a table event.

**Tuple:** ordered set of values. The word originated as an abstraction from the sequence: single, double, triple, quadruple, quintuple, ..., n-tuple.

**UDF:** User-defined function

**UDDI:** Universal Discovery, Description, and Integration

**UDT:** User-defined type (SQL)

**UML:** Unified Modelling Language

**URI:** Uniform Resource Identifier

**URL:** Uniform Resource Locator

**URN:** Uniform Resource Name

**UTC:** Coordinated Universal Time, same as Greenwich Mean Time

**Value expression:** In SQL, an expression which evaluates to a scalar of a known type or a *row value constructor*

**Variance:** Sum of squares of differences from the average, divided by the no. of values

**W3C:** World-wide-web Consortium

**WAR file:** *JAR* that contains a web module

**WSDL:** Web Services Description Language

**XA:** X/Open Distributed Transaction Processing model, providing two-way communication between a transaction manager and resource managers to maintain transaction atomicity while coordinating transactions across applications and application concurrence on shared resources.

**XML:** Extensible Markup Language

**XSL:** Extensible Stylesheet Language

**XSLT:** XSL transformation