_system – SQL keyword. Predefined user, representing the database administrator, who possesses all privileges on all resources of the database. [Atz p137]

100% rule – a rule to help clarify a subclass-superclass relationship. 100% of the superclasses attributes and associations should be applicable to the subclass. See also 'is-a rule' [Lar p399]

1NF – See First Normal Form

2NF – See Second Normal Form

3NF – See Third Normal Form

4NF – See Fourth Normal Form

5NF – See Fifth Normal Form

Absolute – SQL keyword. Used in a 'fetch' statement to move to a particular row of the query. [Atz p141]

Abstract Conceptual Class – a conceptual class is called abstract if every one of its instances must also be an instance of one of its subclasses. See also Concptual Class Partition. [Lar p406]

Access – see Microsoft Access.

Access Manager – part of the DBMS that takes a plan produced by the query optimiser and translates it into accesses to pages of memory containing data. [Atz p321]

Access Method – a software module providing data access and manipulation primitives for each access structure. [Atz p321]

Accessor – a method used to retrieve data from an object. [Atz p405]

Add – SQL keyword: command to add items to a composite object. [Atz p98]

Additive Rule – an inference rule for functional dependencies. Also called the Union Rule. If X → Y and X → Z then X → YZ. Contrast with the projective rule. [Elm p479]

Aggregate Query – an SQL query that returns some function of a collection of rows, rather than the rows themselves. See count,avg,min,max,sum. [Atz p113]

Aggregation – a kind of association used to model whole-part relationships. See also composite, composition, composite aggregation. [Lar p414]

All – SQL keyword. Used in an aggregate query to summarise all non-null values of an attribute, with repetition. Used after union, intersect or except to include/exclude all rows, irrespective of repetition. Used with nested queries to compare an attribute/attributes with the rows returned by the nested query, returning true if all rows satisfy the comparison. [Atz p114,120,123]

All Key relation – a relation whose key consists of all its attributes. [Elm p518]

Alter – SQL keyword: command to change an object specification. [Atz p97]

Analysis Object Model – see Domain Model.

Analysis Patterns – a tool for identifying classes in a conceptual model. [Lar p133]

And – SQL keyword: Used as a Boolean operator to construct the where clause of a query. [Atz p104]

Anomaly – an inconsistency in a database. See Update Anomaly, Insertion Anomaly, Deletion Anomaly, Modification Anomaly. [Elm p471-472]

Any – SQL keyword. Used with nested queries to compare an attribute/attributes with the rows returned by the nested query, returning true if at least one row satisfies the comparison. [Atz p123]

Application Designers – define and create programs for the database. [Atz p9]

Are system generated – SQL-3 keyword. Used with 'values for' to denote that the DBMS should generate OIDs in the given attribute. [Atz p425]

Armstrong's Inference Rules – The additive, augmentation and transitive rules taken together. These three are logically complete, in that any other valid inference rule may be derived from them. [Elm p481]

Array-sequenced organisation – a way data may be arranged in memory (primary or secondary). The sequence of tuples is dictated by an index. [Atz p323]

As – SQL keyword. An optional keyword used to specify aliases for attributes or tables in a query. [Atz p101] SQL keyword: Part of the syntax of 'create view' [Atz p133] SQL-3 keyword: Part of the syntax of 'external' [Atz p426]

Asc – SQL keyword. Used with order by to denote how the rows returned by a query are to be sorted (ascending). [Atz p113]

Assertion – SQL keyword. Used to specify a condition (not dependent on any particular table) that must always be true of the data in the database. [Atz p132,133]

Association – a relationship between (instances of) types that indicates some meaningful and interesting connection. [Lar p153]

Association Class – a class encapsulating (capturing) information about an association between other classes. [Lar p411-413, Bla p285]

Atomic data types – the most simple forms of data, Boolean, string, integer, and so forth. [Atz p399]

Attribute – the name of a column of a table, indicating the meaning of the data in that column. [Atz p18] A logical data value of an object. See simple attribute. [Lar p167]

Attribute Preservation Property – the property of a decomposition D = {R1, R2, … , Rk} of R that the union of the Ri is in fact R. [Elm p503]

Augmentation Rule – an inference rule for functional dependencies. If X → Y then XZ → YZ. Sometimes stated as: If X → Y then XZ → Y. [Elm p479] for MVD's: If X ↠ Y and Z is a subset of W, then WX ↠ YZ. [Elm p516]

Authorization – SQL keyword used to specify the owner of a schema. [Atz p90-91]

Avg – SQL keyword. Used to perform an aggregate query that returns the average value of an attribute. [Atz p115]

Bag – an unordered collection that allows duplicates. [Atz p399]

Bag-of – O2 keyword allowing the construction of bags. [Atz p399]

Base relation – a relation stored in a database, with data that is not calculated from data in other relations. See also derived relation. [Atz p65]

BCNF – see Boyce-Codd Normal Form.

Binary Decomposition – a decomposition of a relation schema into two relation schemas. [Elm p508]

Bit – SQL keyword: domain of binary data. Bits, Booleans, flags, etc. See also varbit. [Atz p88]

Bottom-up design – a design technique that begins with a 'universal relation' containing all interesting attributes, and applies normalization algorithms to create a useful design [Elm p501]

Boyce-Codd Normal Form – a normal form similar to but stronger than the Third Normal Form, often used in its place. [Elm p483] For any nontrivial functional dependency X → A, X must be a superkey (contrast with 'third normal form') [Elm p494]

B-tree – a tree-based structure used for storing data in a database. [Atz p327]

B+ tree – a tree-based structure used for storing data in a database, with extra links to facilitate sequential access to data. [Atz p327]

Bury – Term used by Blaha to describe a strategy for representing an association in an RDBMS, by putting the primary key for one associated class into the table of the other. [Bla p283]

CallableStatement – a class in java.sql used to execute SQL stored procedures. [Ric p182]

Candidate Key – a synonym for key, in a relation schema with more that one key. [Elm p477,485]

Cardinality – the number of elements of the relation (that is, rows in the table) [Atz p17]

Cartesian Product – set of all ordered pairs (or triples etc) of elements from two (or three etc) sets. [Atz p16] A natural join of two relations with no common attributes, whose tuples are all possible combinations of the tuples of the original relations. [Atz p54]

Cascade – SQL keyword. Indicates behaviour of the RDBMS when an object is modified (deleted or changed) when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96] Used in the drop command similarly. See also restrict. [Atz p98] Used similarly in the revoke command to ensure that if a user has passed on privileges to others, their privileges are also revoked when the user's is. See also restrict. [Atz p138]

Cascaded – SQL keyword. Used to indicate that a 'check option' added to a view should also apply to the views used to define it (if any). [Atz p134]

Casual User – a user of the DB who interact with the DB in various, unpredetermined ways. Often familiar with the DDL and DML. [Atz p9]

Char – SQL keyword. Synonym for character. [Atz p88]

Character – SQL keyword: domain of textual data. See char, varchar

Check – SQL keyword. Used in the creation of tables to specify complex conditions the data must satisfy. [Atz p131]

Check Option – SQL keyword. Used to ensure that modifications made to a view result in rows that still belong to the view. [Atz p134]

Class – a container for objects. A class has an interface and an implementation. [Atz p402] In some OODBMSs, a class is just an interface and an implementation, and the object containers are called extensions or extents. [Atz p403]

Class Diagram – a UML notation for a class or concept. [Lar p128]

Class.forName – forName is a method of the Class class in java. It returns an object of type Class representing the named class. A side effect of this is that the class is loaded into the JVM. If the class happens to be a Driver, this in turn will create a Driver object and register the object with the DriverManager. [Ric p184]

Class Hierarchy – aka Generalization-Specialization Class Hierarchy. The way in which subclasses and superclasses are organised, with subclasses placed under their superclasses. See Software Class Hierarchy. [Lar p396]

Client-Server Architecture – a way to organise software using a DBMS and the DBMS itself so that they communicate over a computer network with the DBMS waiting for requests from the software. [Ric p176]

Close – SQL keyword, used to tell the DBMS that the named cursor is no longer needed. Syntax: close *Cursorname*. [Atz p141]

Closure – the closure of a set of functional dependencies F is the set F^+ of all functional dependencies that may be inferred from F. [Elm p481] The closure of a set of attributes X under a set F of functional dependencies is X^+, the set of all attributes Y such that $X \rightarrow Y$ is in F^+. [Elm p481] The closure of a set F of FDs and MVDs is the set of all such that may be inferred from F. [Elm p517]

Coalescence Rule – an inference rule for FDs and MVDs. If $X \twoheadrightarrow Y$ and there exists W such that $W \cap Y$ is empty, $W \rightarrow Z$ and Z is a subset of Y, then $X \rightarrow Z$. [Elm p517]

Column – SQL keyword: used in add, alter or drop to denote a column (attribute) of a table. [Atz p98]

Combination – an unwise strategy for representing a one-to-one association in an RDMBS, by merging the tables of the associated classes. [Bla p284]

Common Associations List – a tool used to identify associations for a domain model – a list of common types of associations. [Lar p155]

Complementation Rule – for MVD's. $\{X \twoheadrightarrow Y\} \models \{X \twoheadrightarrow (R - (X \cup Y))\}$. [Elm p516]

Compilation – the process of optimising a query and converting it into a sequence of page accesses. [Atz p333]

Compiled Query – a query that has been compiled and stored. (see compilation) [Atz p333]

Complete Join – a natural join $r1 \bowtie r2$ in which every tuple of r1 and r2 contributes to the join. [Atz p51]

Complex Data Type – A data type with some detailed structure, for example a record, list, bag or set. [Atz p399]

Composite – the 'whole' in a whole-part relationship (aggregation). [Lar p414]

Composite Aggregation – see Composition. [Lar p415]

Composition – a form of aggregation where the 'part' is part of only one 'whole', and its existence and location and other features are tied to the whole. Contrast with Shared Aggregation. [Lar p415]

Conceptual Class – a category of things in the real world, depicted in a domain model. [Lar p145]

Conceptual Class Category List – a tool to help identify conceptual classes – a list of common categories of conceptual classes. [Lar p134]

Conceptual Class Partition – a division of a conceptual class into disjoint subclasses. [Lar p400]

Conceptual Data Models – any data model in which data is described independently of the logical model used to organise the data, instead relating the data to real-world concepts. An example is the E-R model. Usually used to help design the database. [Atz p6]

Conceptual Model – see Domain Model.

Concurrency – the capacity of a system to handle many users simultaneously. [Atz p417]

connect – a static method of DriverManager used to connect to a database at a specified URL. [Ric p184]

Connection – a class in java.sql that represents a connection between the client software and the database server. [Ric p182]

Constraint – SQL keyword. Used in add, alter or drop to denote a constraint on a table. [Atz p98]

Constraints – SQL keyword. See 'set constraints'.

Constructor – a method used to help create a new object and initialise its data. [Atz p405]

Contradiction – a statement that cannot be true. [Elm p479]

Contravariance – replacing a type by a supertype. See Covariance. [Atz p415]

Control source – Microsoft Access keyword. The property of a form element that indicates the attribute it should display. [Ric p179]

Cost Model – a formula or algorithm for estimating the 'cost' of a particular form of a query. Usually taking into account expected memory usage and time. [Atz p333]

Count – SQL keyword. Used in an aggregate query to count the rows returned. [Atz p114]

Counter – Access SQL keyword. If the data type of an attribute is ‘counter’, then new tuples take on successive integer values. Foreign keys should be of type long. See Identifier domain. [Bla p278]

Covariance – replacing a type by a subtype. See contravariance. [Atz p414]

Cover – a set of functional dependencies F covers another such set E, if E is a subset of the closure of F. [Elm p482]

Create – SQL keyword. Used to create an object in SQL. [Atz p90,91]

createStatement - method of the class Connection used to create a Statement object. [Ric p186]

Current Of – SQL keyword. Used in the where clause of an update or delete statement to indicate that the update or delete should be applied to the current row of the named cursor. [Atz p141]

Cursor – a mechanism in SQL for allowing a set of tuples to be manipulated one by one. [Atz p140]

Cursor for – SQL keyword used to declare a cursor. Syntax: declare *cursorname* [scroll] cursor for *SelectStatement* for <read only | update [of *AttributeList*]> [Atz p140]

Dangling tuple – in an incomplete natural join r1⊳⊲r2, the tuples of r1 and r2 which do not contribute to tuples of the join [Atz p51, Elm p511]

DAO – Data Access Object. One of two ways for Microsoft Access to read data from other databases. See also ‘Link Tables’. [Ric p179]

Data – bits and bytes and strings that have no meaning in and of themselves. [Atz p2]

Data Access Object – see DAO

Database – a collection of data, used to represent information of interest to an information system. [Atz p2] a set of relation (instances) satisfying a given schema [Atz p22]

Database Administrator – Responsible for the design, control and administration of a DB. [Atz p9]

Database Client – a program that connects to a database server in order to use the database controlled by the server. [Ric p176]

Database Instance – a database, in the second sense [Atz p22]

Database Management System – see DBMS

DatabaseMetaData – a class in java.sql that contains information about the database server. [Ric p182]

Database Server – a program that awaits connections from a database client, processes commands from the client, and returns results from the database. [Ric p176]

Data Definition Language – used to define the logical, external and physical schemas and access rights. [Atz p8]

Data dictionary – a collection of information stored in the DBMS about what objects exist. [Atz p99]

Data Independence – the property of a DBMS that allows users and programs to refer to data at a level of abstraction that ignores the actual implementation of the DB. Cf Physical and Logical independence. [Atz p7]

Data Manipulation Language – used to query and update data to the database. [Atz p8]

Data Model – a combination of constructs used to organise data. [Atz p5]

Data type – see simple attribute.

Date – SQL keyword. Domain of date values. See also timestamp. [Atz p89]

Day – SQL keyword. Used to specify Interval attributes. [Atz p90]

DB – see Database.

DBA – see Database Administrator.

DBMS – a software system able to manage large, shared, persistent collections of data while ensuring reliability and privacy. [Atz p3]

DDL – see Data Definition Language.

Deallocate prepare – SQL keyword. Used to indicate that a previously prepared SQL command is no longer needed. Syntax: deallocate prepare *CommandName* [Atz p144]

Decimal – SQL keyword. Synonym for Numeric. [Atz p88]

Decision tree – a tree structure representing the various decisions that need to be made by the query optimiser. See ‘execution plan’ [Atz p340]

Declarativeness – the existence in a system of a high-level query language. [Atz p417]

Declare – SQL keyword. Used to declare cursors. [Atz p140]

Decomposition – a collection D = {R1, R2, …, Rk} of relation schemas that together contain all the attributes of a larger relation schema R. [Elm p502]

Decomposition Algorithm – An algorithm for decomposing a relation into smaller relations that satisfy some normal form. An example is Elmasari Algorithm 15.3. Contrast with 'Relational Synthesis Algorithm' [Elm p509]

Decomposition Rule – See Projective Rule.

Deep Equality – the two objects must have identical values when OIDs in their structure are recursively substituted with the structure of the objects they reference. See superficial equality, identical. [Atz p402]

Deferred – SQL keyword. Used with 'set constraints' to specify that a constraint should only be checked after a full transaction is completed. [Atz p133]

Delete – SQL keyword. Denotes the privilege of being able to delete rows from a table or view. [Atz p137]

Delete From – SQL keyword. Used o remove rows from a table. [Atz p129]

Default – SQL keyword. Used to specify the default value of an attribute or domain. [Atz p92] Used with update…set to change the value of an attribute to the default value. [Atz p130]

Degree (of a relation) – the number of terms in the cartesian product. [Atz p17]

Deletion Anomaly – an inconsistency introduced into a badly designed database when data is deleted without taking into account the bad design. [Elm p472]

Denormalization – the process of transforming a database schema into one satisfying only a lower normal form, usually by storing joins of tables directly instead of as views, for performance reasons. [Elm p484]

Dependency – a dependency between two package exists if one package references elements of the other. [Lar p424] See also 'functional dependency'.

Dependecy Preservation Property – a desirable property of a database schema that all functional dependencies are represented within some individual relations within the schema. [Elm p484] More technically, a decomposition D = {R1, R2, …, Rk} of R has the dependency preservation property with respect to F if the closure of the union of all the $\pi_{Ri}(F)$ equals the closure of F. [Elm p504]

Dependent class – a class whose objects cannot exist without some other objects also existing. See 'Flow of Identity' [Bla p277]

Dereferencing – accessing data in an object referenced by an attribute. [Atz p427]

Derived Attribute – an attribute of a class which may be derived from other attributes or from the nature of associations between objects of the class and other objects. [Lar p175]

Derived Element – an element (attribute or association) that may be derived from other elements in the domain model. [Lar p421]

Derived Relation – a relation which is calculated from other relations in the database. See also base relation, materialized view, virtual relation, view. [Atz p65]

Desc – SQL keyword. Used with 'order by' to denote how the rows returned by a query are to be sorted (descending). [Atz p113]

Design Creep – the (poor) practice of making design and implementation decisions during the analysis phase. [Lar p172]

Destructor – a method used to cancel (destroy) an object, and possibly other linked objects. [Atz p405]

Dictionary – see Data Dictionary.

Distinct – SQL keyword. Used in an aggregate query to summarise distinct non-null values of an attribute. [Atz p114]

Difference (between two relations) – the difference r1 – r2 is the set of all tuples belonging to r1 but not to r2. [Atz p43]

Distinct – SQL keyword: 'select distinct *attr*' causes the query to discard duplicate rows in the returned view. [Atz p108]

Distinct Table – a phrase used in Blaha to indicate that an association should be represented by a table in the RDBMS distinct from the tables for the classes. [Bla p282,283,etc]

DML – see Data Manipulation Language.

Domain (of a relation) – one of the sets used to form the Cartesian product of which the relation is a subset. That is, the type of data that appears in a column of a table. [Atz p16]

Domain – SQL keyword specifying that an operation acts on a domain object. [Atz p91]

Domain Constraint – (also Value constraint). A form of tuple constraint which specifies allowable values of particular attribute (eg Mark must be between 0 to 100). [Atz p29]

Domain Model – a visual representation of conceptual classes of real-world objects in a domain of interest. [Lar p128]

Domain Object Model – see Domain Model.

Double-dot notation – Notation in SQL-3 (..) used to access subcomponents of the object stored in an attribute. [Atz p427]

Double precision – SQL keyword for double precision values. See also float, real. [Atz p89]

Driver – a software component allowing a software system to use databases stored on DBMSs by a particular vendor. [Ric p179] An abstract class (interface??) in the java.sql package that is subclassed by database vendors to provide Drivers for their databases. [Ric p182]

Driver Manager – software that is aware of the available drivers on a system, and is able to load the correct drivers to handle requests as the requests arrive. [Ric p179]

DriverManager – A class in java.sql that provides a Driver Manager. [Ric p182]

Driver Package – a software component that may be incorporated into database client software, allowing it to use a particular driver. [Ric p179]

Drop – SQL keyword. Command to delete an object. [Atz p98]

Durability – The capacity of a system to support persistent data. [Atz p417]

Dynamic Lookup – see late binding.

Dynamic properties – that part of a class or type definition that describes the behaviour of the objects (that is, the methods). [Atz p399]

Dynamic SQL – SQL statements that are generated and used by a program at runtime rather than compile time. [Atz p142]

Elementary Domain – the basic domains (datatypes) available in SQL. [Atz p88]

Elimination – a strategy for representing single inheritance in an RDBMS by simply ignoring subclasses with no extra attributes. [Bla p288]

Embedded SQL – SQL statements inserted into a program written in a general-purpose programming language. The program must be processed by a pre-processor before being compiled. [Ric p181]

Encapsulation – the ability, in a system, to hide data inside objects and only allow access to the data via public methods. [Atz p416] Inbuilding SQL commands into a program written in a general purpose computer language. [Atz p139]

End User – uses the database in fixed, routine, predefined ways. [Atz p9]

Entity-Relationship Model – An example of a conceptual data model. [Atz p6]

Entry-sequenced organisation – a way data may be arranged in memory (primary or secondary). The sequence of tuples is dictated by the order of entry. [Atz p323]

Entry SQL – basic level SQL implemented in all commercial RDBMSs [Atz p86]

Enumeration Domain – a data type allowing a few fixed values. Eg, ("Mon", "Tue", … , "Sun"). [Bla p278]

Enumeration Encoding – a correspondence of enumeration values to numbers. [Bla p280]

Enumeration String – A string corresponding to a value of an enumeration domain. [Bla p279]

Enumeration Table – a table in an RDBMS containing the allowed values of an enumeration. [Bla p280]

Equality – see superficial equality and deep equality. [Atz p402]

Equi-join – a theta join where the tuples of the Cartesian product are selected according to a number of equalities between attributes. [Atz p54]

Equivalent – two sets E and F of functional dependencies are equivalent if their closures are equal. [Elm p482]

E-R model – see Entity-Relationship model.

Except – SQL keyword. Used to find the 'difference' of the output of two SQL 'select' statements (queries). [Atz p120]

EXEC SQL – Typical keyword used to embed SQL statements in a general purpose programming language. [Ric p181]

Execute – SQL keyword. Used to execute a previously prepared SQL command. Contrast with 'execute immediate'. Syntax: execute *CommandName* [into *TargetList*] [using *ParameterList*] [Atz p144]

execute – a method of the class statement that may be used to send any SQL command to a database server. [Ric p186]

Execute immediate – SQL keyword. Causes the immediate execution of an SQL command contained in a given string. Syntax: execute immediate *SQLString*. Used to make dynamic SQL possible. [Atz p143]

executeQuery – a method of the class Statement used to send a SELECT statement to a database server. [Ric p186]

executeUpdate – a method of the class Statement used to send any statement *except* a SELECT statement to a database server. [Ric p186]

Execution plan – a set of choices that might be made by a query optimiser. Represented by a leaf node of a decision tree, that is, by a path through the decision tree. [Atz p340]

Existence-Based Identity – Objects are distinguished via an object identifier, that is, in the RDBMS, an extra field is added to each table to contain the ID of the object. [Bla p276]

Exists – SQL keyword. Used with nested queries. Exists (*Query*) returns true if *Query* returns at least one row. [Atz p126]

Extensibility – the ability to define new types or classes based on existing ones. [Atz p416]

Extension – The set of examples of a conceptual class [Lar p131] See 'class' [Atz p403] A particular instance of a relation schema [Elm p477]

Extensional Component (of a database) – the state or instance. [Atz p6]

Extent – see 'class' [Atz p403]

External – SQL-3 keyword. Used as part of a 'returns' statement to indicate thyat a function is defined outside the DBMS in a general purpose language. Syntax: 'returns *type* as external name *filename* language *language*' [Atz p426]

External Schema – a particular view of the database as presented to a particular user. [Atz p7]

Fat Relation – a relation with many attributes. [Elm p473]

Fetch – SQL keyword. Used to retrieve data from a query via a cursor. Syntax: fetch [*Position* from] *Cursorname* into *FetchList*. [Atz p140]

Fifth Normal Form – The information content of the database cannot be reconstructed from smaller record types, that is, record types with smaller numbers of attributes than the original. [Ken p123]

First – SQL keyword. Used in 'fetch' to retrieve the first row of the query. [Atz p140]

First Normal Form – all occurrences of a record must contain the same number of fields. [Ken p120] Data types of attributes must be atomic, and relations may not be nested. [Elm p485]

Float – SQL keyword. Domain of floating point values. See also double precision, real. [Atz p89]

Flow of Identity – notation added to the UML diagram of the domain model indicating which objects or classes derive identity (existence etc) from which others. Identity flow is indicated with an arrow from a dependent class to an independent class. See independent class, dependent class. [Bla p277]

From – SQL keyword. Used to construct queries. Specifies what tables the attributes are selected from. [Atz p101] SQL keyword, used to specify the user in a ‘revoke’ statement. [Atz p138] SQL keyword used if a position is specified when ‘fetch’ing data from a cursor. [Atz p140]

For – SQL keyword. Part of the syntax for ‘cursor’ declaration. [Atz p140]

Foreign key – SQL keyword. Used to define a referential constraint in SQL. See also ‘references’. [Atz p96]

Foreign Key Constraint – see Referential constraint.

Foreign Key Attribute – a (poor choice) attribute which is a piece of text or similar basic data type which actually refers to a complex object. Should be removed and replaced with an association. [Lar p172]

Form – a page of a Graphical User Interface used to display the results of a query. [Ric p178]

Fourth Generation Language – a highly sophisticated development tool allowing easy creation of database management applications. [Atz p139]

Fourth Normal Form – a record type should satisfy 3NF, and should not contain more than one independent `mutivalued fact’ about a key. [Ken p122] a relation R in the fourth normal form with respect to a set of MVDs F, if for every nontrivial $X \twoheadrightarrow Y$ in F^+, X is a superkey of R. [Elm p517]

Full functional dependency – a functional dependency $X \to Y$ where $X' \to Y$ does *not* hold for any proper subset X’ of X. See also ‘partial dependency’, ‘second normal form’ [Elm p488]

Full outer join – an outer join $r1 \rhd\lhd_{\text{FULL}} r2$ where dangling tuples from both r1 and r2 are padded with blanks and inserted into the join. [Atz p53]

Full [Outer] Join – SQL keyword: used to join two tables before selecting from them. See join. [Atz p109]

Full SQL – implementing even the most advanced and newest features of the SQL standard. [Atz p86]

Function – SQL-3 keyword used to manipulate (create etc) functions.

Functional Dependency – a field (or set of fields) A is functionally dependent on a set of fields X if it is invalid to have two records with the same values for X but different values for A. [Ken p121] More formally, Y is functionally dependent on X if for any two tuples t1, t2 in the universal relation schema, we have $t1[X] = t2[X]$ implies $t1[Y] = t2[Y]$. Denoted $X \to Y$.

Functions – terminology used in SQL-3 for methods. [Atz p426]

Generalization – identifying commonality among concepts and defining a more general concept encompassing them. See superclass, specialization. [Lar p396] A form of migration where an object moves from a subclass to a superclass. [Atz p410]

Generalization-Specialization Class Hierarchy – see class hierarchy. [Lar p396]

Generalization Table – a table in an RDBMS used to indicate which classes are subclasses of which superclasses. [Bla p291]

getColumnCount – a method of ResultSetMetaData that returns the number of columns of the ResultSet. [Ric p190]

getColmunName – a method of ResultSetMetaData that returns the name of a given column. [Ric p190]

getColumnType – a method of ResultSetMetaData that returns a code representing the type of data stored in a given column. [Ric p190]

getColumnTypeName – a method of ResultSetMetaData returning a String representing the type of data stored in a given column. [Ric p190]

getMetaData – a method applied to a ResultSet object that returns the ResultSetMetaData associated with the ResultSet. [Ric p187]

getResultSet – a method of Statement to retrieve the most recently generated ResultSet. Usually used after an execute statement was performed with a SELECT query as the argument. [Ric p195]

getString – a method of ResultSet that returns the data in the current row of the given column. See also next. Methods getFloat, getInt, getBigDecimal etc also exist (see box on Ric p191). [Ric p190]

getTimeDateFunctions – a method applied to a DatabaseMetaData object to discover the time and date functions available on the database server. [Ric p186]

getUpdateCount – a method of Statement used to determine the number of rows affected by the most recent update or delete query. Usually used after the execute method is used to send on non-select SQL command to the database. [Ric p195]

getUserName – a method applied to a DatabaseMetaData object to discover the username. [Ric p186]

Grant – SQL keyword. Used to give a privilege on a resource to a user. Syntax: grant *priv* on *res* to *user* [with grant option]. [Atz p137]

Group by – SQL keyword. Used to modify an aggregate query to partition the rows according to the values of given attributes before doing the calculations required by the aggregate query. See [Atz p116]

Hash-based structure – data is arranged in memory via a hash table structure. [Atz p325]

Hash join – a 'join method' where a hash function is used to identify matching tuples in the two tables. [Atz p338]

Having – SQL keyword. Similar to 'where', it specifies a Boolean condition that must be satisfied by the rows finally returned from an aggregate query with a 'group by' clause. [Atz p118]

Hierarchical Data Model – a data model using tree structures to organise data. [Atz p5]

Host Language – a high-level language in which DDL or DML commands may be embedded. [Atz p8]

Hour – SQL keyword. Used to specify Interval attributes. [Atz p90]

Identical – objects in an OODBMS such as O2 are identical if they share the same OID. See also superficial equality, deep equality. [Atz p402]

Identifier Domain – an RDBMS specific domain allowing easy allocation of new object identifiers. See Counter, Sequence. [Bla p278]

Identity – the manner in which individual objects are distinguished in an RDBMS. See Value-Based Identity, Existence-Based Identity. [Bla p276]

Immediate – SQL keyword. Used with 'set constraints' to indicate that a given constraint should be immediately checked after every step of a transaction, not merely when the whole transaction is completed. [Atz p133]

Impedance Mismatch – The fact that an SQL query returns whole blocks of data, but high-level general-purpose languages generally can only handle single items of data one at a time – and the problem of using the two approaches together. [Atz p407, p139]

Implementation (of a class) – the implementation of the methods of a class. Sometimes also descriptions of the data structures to be used for storing its data. [Atz p403]

Implementation (of a method) – See 'method implementation'.

In – SQL keyword. Used with nested queries. *Attr* in (*Query*) is equivalent to *Attr* = any (*Query*). [Atz p125]

Incomplete join – a natural join r1⊳⊲r2 where *not* every tuple of r1 and r2 contributes to a tuple of the join. See: dangling tuple. [Atz p51]

Independent class – a class whose objects exist "independently", that is, they do not depend on other classes for their existence or identity. See 'Flow of Identity' [Bla p277]

Independent multivalued fact – two multivalued facts are independent if the actual values for one fact do not affect the possible or actual values of another. See 'fourth normal form' [Ken p122]

Index – a structure (usually a tree structure) allowing quick access to data stored in the database via a key. [Atz p337]

index – SQL keyword. Used to create or drop an index for a database. Syntax: 'create [unique] index *IndexName* on *TableName*(*AttributeList*)' or 'drop index *IndexName*'. [Atz p343]

Indexed access – access to data via an index. [Atz p337]

Indivisible Data Type – a synonym for atomic data type. [Elm p485]

Infer – deduce logically. Used in Elmasari to refer specifically to functional dependencies. A functional dependency X → Y can be inferred from a set F of functional dependency if X → Y can be deduced logically from F via the application of various inference rules. [Elm p479]

Inference Rules – a rule that may be used to infer functional dependencies from others. See Reflexive Rule, Augmentation Rule, Transitive Rule, Projective Rule, Additive Rule, Pseudotransitive rule. [Elm p479]

Information – data within a context that provides it with meaning. [Atz p2]

Information system – the procedures whereby information within an organization is managed. [Atz p1]

Inheritance – a software mechanism allowing superclass 'things' (associations, attributes and methods) to apply to subclasses. [Lar p409]

Inner Join – SQL keyword: used to join two tables before selecting from them. See join. [Atz p109]

Input parameter – data that is received by a method at the time it is called. [Atz p405]

Insert – SQL keyword. Denotes the privilege of being able to add data to a table or view. [Atz p137]

Insert into – SQL keyword. Used to add data to a table. [Atz p128]

Insertion Anomaly – an inconsistency introduced into a badly designed database when new data is inserted without taking into account the bad design. [Elm p471]

Instance (of a database) – the values stored within the database at a particular time. [Atz p6] (of a class) an object that can belong to the class, but cannot belong to any of its subclasses without modification. Contrasted with 'member'ship of objects. [Atz p410] Synonym for Extension [Elm p477]

Integer – SQL keyword. Domain of integers. See also numeric, decimal, smallint. [Atz p88]

Integrity Constraint – a property that must be satisfied by all correct database instances. See Predicate, intra-relational constraint, inter-relational constraint. [Atz p29]

Intension – the definition of a conceptual class [Lar p131]

Intensional Component (of a database) – the schema. [Atz p6]

Interface – the 'type' of a class. A description of its data and method signatures. [Atz p402]

Intermediate SQL – features of SQL that are commonly used in commercial products. [Atz p86]

Internal Schema – the implementation of the logical schema by means of physical storage structures (files, etc). [Atz p7]

Inter-record redundancy – a redundancy in the database schema that does not arise within a single relation taken alone. [Ken p125]

Inter-relational constraint – an integrity constraint that involves more than one relation of the database (eg StudentNumber in EXAMS must match a value of StudentNumber in STUDENTS). Contrast with Intra-relational constraint. [Atz p29]

Intersect – SQL keyword. Used to find the intersection of the output o two select statements (queries). [Atz p120]

Intersection – the intersection of two relations r1 and r2 is the set of tuples belonging to both r1 and r2 (contrast with union or difference). [Atz p43]

Interval – SQL keyword, domain of time intervals. [Atz p89]

Into – SQL keyword. Part of the syntax of 'fetch'. [Atz p140] SQL keyword. Part of the syntax of 'execute'. [Atz p144]

Intra-Relational Constraint – a form of integrity constraint that is defined with reference to a single relation or table in the database. Examples are tuple constraints and domain or value constraints. Contrast with Inter-relational constraint. [Atz p29]

Invocation – see 'method invocation'.

Is – SQL keyword: used in the where clause of a query: “is null” or “is not null”. [Atz p106] SQL-3 keyword: used as part of the ‘scope for’ syntax. [Atz p425]

Is-a rule – a rule to help clarify a subclass-superclass relationship. Instances of a subclass must also be instances of a superclass. [Lar p400]

Java Database Connectivity – see JDBC.

Java.sql – A standard package in java allowing use of JDBC. [Ric p180]

JDBC – Java Database Connectivity. A method, similar to ODBC, for allowing java programs to connect to database servers. [Ric p180]

JDBC-ODBC bridge – software that is able to translate commands sent to JDBC and forward them on to an ODBC driver. [Ric p182]

Join – An operator combining tuples from different tables (relations) to form a new table (relation) see natural join, outer join, theta-join, equi-join, Cartesian product (2nd meaning). [Atz p49-56]

Join – SQL keyword: used to join two tables before selecting from them. See inner join, left join, right join, full join. [Atz p109]

Join Method – a method the DBMS can use to accomplish a join. See ‘nested loop’, ‘merge scan’, ‘hash join’. [Atz p337]

Key – a minimal superkey. That is, a set of attributes A on a relation r, such that there is no two distinct tuples t1 and t2 of r with t1[A] = t2[A]… and A does not contain any proper subset for which this statement still holds. [Atz p31] A set of attributes identified by a key constraint. [Atz p30-33]

Key Constraint – an (intra-relational) integrity constraint ensuring that a selected set of attributes forms a (super)key. [Atz p30-33]

Language – SQL-3 keyword. Used as part of the syntax for ‘external’ to indicate what language the external function is written in. [Atz p426]

Last - SQL keyword. Used in a ‘fetch’ statement to retrieve the last row of a query. [Atz p140]

Late Binding – a feature of an OO system, where the method to be used is only decided at runtime, not at compile time. Allows method overloading and overriding to work nicely. Also called ‘dynamic lookup’ [Atz p413]

Left outer join – an outer join r1 $\rhd\!\lhd_{\text{LEFT}}$ r2 where dangling tuples from r1 are padded with blanks and inserted into the join. [Atz p53]

Left [Outer] Join – SQL keyword: used to join two tables before selecting from them. See join. [Atz p109]

Legal Extension – an extension of a relation schema that satisfies the (functional dependency) constraints. Also called Legal Relation State. [Elm p477]

Legal Relation State – see 'Legal Extension'.

Like – SQL keyword: Used as an operator to construct the where clause of a query. Specifies that a string attribute must match (be like) a certain pattern. [Atz p105

Link Attribute – An attribute of an association. [Bla p285]

Link Tables – a menu item in Microsoft Access allowing the access database to gain access to external sources of data. See also DAO. [Ric p179]

List – an ordered collection of objects of the same type, allowing duplicates. [Atz p399]

List-of – O2 keyword allowing the construction of lists. [Atz p399]

Local – SQL keyword. Used to indicate that a 'check option' on a view should only apply to the view itself, not to pther views that were used to define it. [Atz p134]

Logical Data Model – any data model where a particular method of organisation is used to organise data. Examples are the Relational, Hierarchical, Network and Object data models. [Atz p6]

Logical Independence – (see Data Independence) the interaction with the data does not depend on the logical arrangement of data within the database. [Atz p8]

Logical Schema – a description of a database according to the appropriate logical data model. [Atz p7]

Lossless Join Property – a vital property of a database schema, that joins do not introduce spurious tuples. Also called the non-additive join property. [Elm p484, 505]

Lost – some functional dependencies are **lost** if the dependency preservation property is not satisfied by a decomposition. [Elm p504]

Many-to-Many association – an association between two classes A and B, where each A can be associated with many B's, and vice versa. See Multiplicity. [Bla p282]

Materialized view – a derived relation whose data is actually stored in the database. Contrast with virtual relation (view) [Atz p65]

Max – SQL keyword. Used to perform an aggregate query that returns the maximum value of an attribute. [Atz p114]

Merge-scan – a 'join method' useful when both tables are pre-sorted on the join attributes. [Atz p338]

Member (of a class) – an object contained in the class. The object will also simultaneously be members of all the classes superclasses. Contrasted with ‘instance’ [Atz p410]

Metadata – data about the structure of data. See reflexivity, dictionary. [Atz p99]

Method – an action that may be performed on an object. [Atz p404]

Method Body – See method implementation.

Method Implementation - the (code describing the) operations that a method performs. [Atz p403,404]

Method Invocation – the act of performing the action on the object. May be seen as an act of sending a message to an object. [Atz p406]

Method Overloading – the situation where the same method name (but different signatures) is used more than once in a given class. [Atz p413]

Method Overriding – the situation where a method defined in a superclass is redefined (with the same signature, but different implementation) in a subclass. [Atz p412]

Method Signature – a description of the parameter types and return type of a method. [Atz p402,404]

Microsoft Access – an example of a database development tool, produced by Microsoft. [Ric p178]

Microsoft Jet – a driver for a Microsoft Access database. [Ric p179]

Middleware – Software that waits for requests from database clients, reinterprets the requests, and passes them on to a DBMS. [Ric p176]

Migration – the event of an object moving from a superclass to a subclass (specialization) or vice-versa (generalization). [Atz p409]

Min – SQL keyword. Used to perform an aggregate query that returns the minimum value of an attribute. [Atz p114]

Minimal – a set F of function dependencies is minimal if (i) Every function dependency in F is of the form $X \rightarrow A$ where A is a single attribute, (ii) If any $X \rightarrow A$ in F is replaced by $Y \rightarrow A$, where Y is a proper subset of X, the new set of functional dependencies is not equivalent to F, and (iii) if any $X \rightarrow A$ is removed from F, the new set of functional dependencies is not equivalent to F. [Elm p482]

Minimal Cover – a minimal cover F_{min} for a set of functional dependencies F is a minimal set of functional dependencies that covers F. [Elm p482]

Minute – SQL keyword. Used to specify Interval attributes. [Atz p90]

Modification Anomaly – an inconsistency introduced into a badly designed database when data values are changed without taking into account the bad design. [Elm p472]

Month – SQL keyword. Used to specify Interval attributes. [Atz p90]

Multidetermine – X multidetermines Y means $X \twoheadrightarrow Y$. (see MVD). [Elm p514]

Multiple Inheritance – the situation where a class is a subclass of more than one superclass. [Bla p291, Atz p410]

Multiplicity – characteristic of a role. It indicates how many objects of one type fulfil the role for the object at the other end of the association. [Lar p157]

Multi-target method – a method which can be applied to any of a number of objects. [Atz p404]

Multivalued Domain – a data type where the variable may take on more than one value at a time. [Bla p282]

Multivalued Dependency – see MVD.

Multivalued Fact – information about something that may take on more than one value simultaneously. For example, InvoiceNumber for Customer. See 'independent multivalued fact', 'fourth normal form'. [Ken p122]

MVD – multivalued dependency. Occurs when two or more independent multivalued facts are stored in a single table. Formally, a multivalued dependency $X \twoheadrightarrow Y$ specifies the following constraint: if tuples t1 and t2 exist with $t1[X] = t2[X]$, then there should also exist tuples t3, t4 with (*) $t3[X] = t4[X] = t1[X]$ ($=t2[X]$), (*) $t3[Y] = t1[Y]$ and $t4[Y]=t2[Y]$, (*) $t3[Z] = t2[Z]$ and $t4[Z] = t1[Z]$ (where $Z = R – (X \cup Y)$). [Elm p514]

Name – SQL-3 keyword. Part of the syntax for 'external'. [Atz p426]

Name Conflicts – a problem with multiple inheritance, where two superclasses of a class use the same name for different properties. [Atz p411]

Natural Join – an operator combining tuples of two relations r1 and r2 on sets of attributes X1 and X2. The new tuples are formed on $X1 \cup X2$ by combining tuples from r1 and r2 for which $X1 \cap X2$ match. $r1 \bowtie r2 = \{t \text{ on } X1X2 : t[X1] \in r1 \text{ and } t[X2] \in r2\}$. See complete and incomplete join. [Atz p49]

Nested Loop – a 'join method' where the attributes of one table are looped through once for each tuple in the other. [Atz p337]

Nested Query – A select statement (SQL query) used as part of the with clause of another query, and used as a source of data against which to compare attributes. [Atz p122-128]

Nested Relation – a relation stored within the tuples of another relation. [Elm p487]

Nesting – in SQL-3 – taking data on the same level, and (for example by means of the 'set' aggregate operator) returning a more multi-levelled data structure from a query. [Atz p428]

Network Data Model – A data model using graphs to organise data [Atz p5]

New – an O2 keyword used to invoke the method that creates an object. [Atz p406]

next – a method of ResultSet that moves the ResultSet's (inbuilt) cursor to the next row of the table (or to the first row when next is called for the first time) It returns true if successful, or false if all rows of the ResultSet have already been read.

Next – SQL keyword. Used in a 'fetch' statement to retrieve the next row of a query. [Atz p140]

Nil – O2 keyword for the null value. [Atz p399]

No action – SQL keyword. Indicates behaviour of the RDBMS when an object is modified (deleted or changed) when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96]

No-information – a null value indicating that it is not known whether information exists, and even if it does, the value is unknown. Eg. "Phone Number" for a person who may or may not have a phone. In practice the situation faced by users of databases with null values. [Atz p23]

Non-additive join property – see Lossless Join Property.

Non-existent value – a null value indicating that the information does not exist, for example "Phone Number" for a person who has no phone. [Atz p23]

Nonprime attribute – an attribute which does not form part of any candidate key. Contrast with 'prime attribute'. [Elm p485]

Nontrivial functional dependency – a functional dependency $X \rightarrow Y$ where Y is not a subset of X. [Elm p479]

Nontrivial MVD – a MVD which is not trivial. That is, $X \twoheadrightarrow Y$ where Y is not a subset of X, nor is $X \cup Y$ equal to R. [Elm p516]

Normal Form – a convention for good database design. See First Normal Form, Second Normal Form, etc up to Fifth Normal Form. Also Boyce-Codd Normal Form. [Ken p120] Also: The Normal Form of a relation is the highest normal form it satisfies. [Elm p484]

Normalization – the process of changing a database design to comply with the various normal forms. [Ken p121]

Normalization Algorithm – an algorithm for taking an unnormalized relation and putting it into a higher normal form. [Elm p501]

Not exists – SQL keyword. Used with nested queries. Not exists (*Query*) returns true if *Query* returns no rows at all. [Atz p126]

Not in – SQL keyword. Used with nested queries. *Attr* not in (*Query*) is equivalent to *Attr* <> all (*Query*). [Atz p127]

Not null – SQL keyword. Constraint that the given attribute may not be null. [Atz p93] Part of a where clause of a query. See 'is'. [Atz p106]

Noun-phrase analysis – a tool for identifying concepts, where noun phrases in the requirements documents are sought. [Lar p135]

Null – SQL keyword indicating a null value. See also 'not null' [Atz p92] Part of a where clause of a query. See 'is' [Atz p106] Used with update…set to change the value of an attribute to null. [Atz p130]

Null Value – a special value a tuple can assume on an attribute, denoting an absence of information. See Unknown Value, Non-existent Value and No-information. [Atz p26]

Numeric – SQL keyword: domain of exact numbers, either integral or with a given number of decimal places. See also decimal. [Atz p88]

O2 – a specific example of an OODBMS. [Atz p299]

Object Data Model – A data model encapsulating an object-oriented approach. [Atz p5]

Object Identifier – see OID.

Object Identity – the property of OODBMSs that Objects have OIDs. [Atz p416]

Object-Oriented Analysis – an analysis of a problem domain that divides it according to objects and classes, that is, things and types. [Lar p132]

Object-Valued – an object valued property is a property whose value is an instance of an object, that is, an OID of an object. [Atz p401]

ODBC – 'Other Database Connectivity'. A standard protocol that allows applications to communicate with databases without having to know in advance which particular database server products it will need to connect to. [Ric p179]

Of – SQL keyword. Part of the syntax used to declare (create) cursors. See ‘cursor for’ [Atz p140]

Of Type - SQL-3 keyword used to define a table whose rows are of a predefined tuple type. [Atz p423]

OID – Object Identifier. A key used to uniquely identify an object. [Atz p399]

OODBMS – Object-Oriented DBMS. [Atz p398]

On – SQL keyword. See On delete, On update [Atz p96]. SQL keyword: Used to specify conditions on how a join should be performed [Atz p109] Used to specify the resource in a grant or revoke statement. [Atz p137,138] Used to specify the table and attributes in ‘create index’. [Atz p343]

On Delete – SQL keyword. The beginning of a clause indicating the behaviour of the RDBMS when an object is deleted when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96]

One-to-Many Association – an association between two classes A and B, where each A may be associated with many B’s, but each B is only associated with one A. See Multiplicity. [Bla p283]

One-to-One Association – an association between two classes A and B, where each A is associated with only one B, and vice-versa. See Multiplicity. [Bla p283]

On Update – SQL keyword. The beginning of a clause indicating the behaviour of the RDBMS when an object is changed when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96]

Open – SQL keyword. Used to run the query associated with a cursor, and begin to manipulate the data. See ‘cursor’. Syntax: open *cursorname*. [Atz p140]

Optimisation – the process of choosing “the best” of several equivalent forms of a query, for example (in ‘cost-based’ optimisation) choosing the form with the lowest estimated cost (see ‘cost model’). [Atz p333]

Or – SQL keyword: Used as a Boolean operator to construct the where clause of a query. [Atz p105]

ORDBMS – Object-Relational DBMS. An RDBMS with some Object-Oriented concepts added. An example is SQL-3. [Atz p423]

Order by – SQL keyword: Used to sort the output of a query. See asc, desc. [Atz p113]

Ordered – UML keyword: Used to denote that objects associated with another are to be kept in some order. [Lar p423]

Ordered Association – an association where the Ordered keyword is applied. [Bla p286]

Orthogonal – the constructors for various data types in O2 and SQL are ‘orthogonal’, meaning they can be arbitrarily nested. [Atz p399]

Other Database Connectivity – see ODBC.

Outer Join – a natural join augmented with tuples derived from tuples of r1 or r2 for which no matching tuple in the other relation exists. New tuples are padded with blanks for the missing values. See left outer join, right outer join and full outer join. [Atz p53]

Output parameter – the data that a method returns back to the caller. That is, the return value. [Atz p405]

Overloading – see Method Overloading.

Overriding – see Method Overriding.

Ownership – a package ‘owns’ a class if the package contains the class. [Lar p424]

Package – used to group together classes which are similar or related in some way, to ease the software development process. [Lar p424-425]

Page – a block of memory. [Atz p321]

Parameter – see Input parameter, Output parameter. [Atz p405]

Partial Dependency – a functional dependency X → Y where X’ → Y also holds for some proper subset X’ of X. See also ‘full functional dependency’ [Elm p488]

Partition – see Conceptual Class Partition.

Persistent – the lifespan of a database extends beyond that of the program using it. [Atz p4] The lifespan of an object extends beyond the execution of the program using it. [Atz p411-412]

Pipelining – a memory-saving technique of performing several operations tuple by tuple, and so not storing intermediate tables. [Atz p341]

Physical Independence – (see Data Independence) the interaction with the DBMS does not depend on how the data is physically stored on the disk (or elsewhere). [Atz p7]

PL/SQL – an extension of SQL marketed by Oracle. [Atz p146]

Pointer-based model – a logical data model where pointers are used to refer to data in different parts of the database. [Atz p21]

Polymorphic value – a value that may belong to any of a number of types. An example is ‘nil’ (the null value). [Atz p399]

Predicate – a function associating a value True or False with an instance of a database [Atz p29]

Prepare - SQL keyword. Used to prepare an SQL command from a string for later use. Syntax: prepare *CommandName* from *SQLString*. See also ‘deallocate prepare’ [Atz p144]

PreparedStatement – a class in java.sql that represents a precompiled SQL command. [Ric p182]

Primary Key – a key (in the second sense) that is constrained to not contain null values [Atz p33] An arbitrarily selected Candidate Key used for identification of tuples in a relation. [Elm p485]

Primary key – SQL keyword. A constraint that the given attribute(s) form the primary key of the table. [Atz p94]

Prime Attribute – an attribute which is a member of some candidate key. Contrast with ‘nonprime attribute’ [Elm p485]

Privacy – Each user is qualified to perform only certain actions on the database. [Atz p4]

Private method – a method that may only be called by other methods of the same class. See ‘public method’ [Atz p405]

Privilege – a ‘permission’ to do something on some component of a database (some ‘resource’) [Atz p136]

Procedure – SQL keyword. Used to define a procedure. Under standard SQL, a procedure may only contain a single SQL statement. Many DBMSs relax this restriction. [Atz p145]

Projection – An operator that takes a relation and returns a new relation whose attributes are a subset of the original. [Atz p47-48] The projection of a set F of functional dependencies onto a relation schema R is the set of all X → Y in F^+ such that X and Y are subsets of R. Denoted $\pi_R(F)$. [Elm p503]

Projective Rule – An inference rule for functional dependencies. Also called Decomposition Rule. If X → YZ then X → Y (and X → Z). [Elm p479]

Proof by Contradiction – a method of proof where one assumes the opposite of the thing one is trying to prove, and deduces a contradiction. [Elm p479]

Properties – The properties of an object are the data members that make up the object, that is, the attributes of the type of the object. [Atz p400]

Pseudotransitive Rule – an inference rule for functional dependencies. If X → Y and WY → Z, then WX → Z.

Public method – a method that is accessible by any component of the system. See ‘private method’ [Atz p405]

Push (up or down) – a term used by Blaha to indicate a strategy where a subclass or a superclass is not represented in the RDBMS, and its attributes are stored in the superclass or subclass table instead. [Bla p290]

Qualified Association – an association with a qualifier. [Lar p422, Bla p285]

Qualifier – information added to an association in a UML diagram to give more information on how an association is implemented. [Lar p422]

Query – a function mapping instances of a given database schema into relations on a given set of attributes. [Atz p56]

Query Language – a language in which queries may be expressed. [Atz p56]

Read only – SQL keyword. Used in the creation of a cursor to indicate that the data returned by the query may not be modified. [Atz p140]

Real – SQL keyword. Domain of lower precision values. See double precision, float. [Atz p89]

Record – an ordered sequence of values of possibly different types. [Atz p399]

Record-of – O2 keyword allowing the construction of records. [Atz p399]

Record source – Microsoft Access keyword. The property of a form that indicates the source of the data it displays. [Ric p178]

Ref – SQL-3 keyword. Used to define an attribute to have values that reference a particular type of tuple. [Atz p424]

References – SQL keyword: specifying a referential constraint. The values of the given attribute(s) match values of other attributes in other table. [Atz p95] SQL keyword denoting the privilege of being able to use a table or attribute as a foreign key in ones own tables. [Atz p137] In Object-Oriented Analysis: a class references another if it uses or is associated with it. A package references a class if any of its classes do. See dependency. [Lar p424]

Referential Constraint – (also: foreign key constraint). A constraint ensuring, for a set of attributes A of a relation r1, and a corresponding set of attributes B of r2, and is a key (the primary key?) for r2, that for every tuple t1 of r1, there exists a tuple t2 of r2 for which t1[A] = t2[B]. [Atz p34,35]

Refinement – properties and a method's input parameters can be refined by giving them new types which are subtypes (or in some cases, supertypes) of the original types. See 'covariance' and 'contravariance' [Atz p414]

Reflexive Association – an association of a concept with itself. [Lar p423]

Reflexive Rule – An inference rule for functional dependencies. If Y is a subset of X, then Y is functionally dependent on X. An alternative (but not equivalent) definition: X is functionally dependent on X. [Elm p479]

Reflexivity – the property of a DBMS that it can store, as data, data about data. See Metadata, Dictionary. [Atz p99]

Relation – a subset of a Cartesian product [Atz p16] More usually, in database theory, a collection of tuples [Atz p19,22]

Relation Instance – a relation, in the second sense. [Atz p22]

Relational Data Model – a data model using tables (relations) to organise data. [Atz p5, 15]

Relational Synthesis Algorithm – an algorithm for decomposing a (universal) relation schema into a decomposition with the dependency preservation property. Examples are algorithms 15.1 and 15.4 of Elmasari. Contrast with Decomposition Algorithm. [Elm p504, 505, 509]

Relation Profile – statistical information about a relation, such as the number of tuples, the number of bytes per tuple, etc. Used in cost-based optimisation to estimate the cost. [Atz p333-334]

Relation Schema – the name of the relation R, and a set X of names of the attributes. Normally denoted R(X). [Atz p22]

Relative – SQL keyword. Used in a 'fetch' statement to move a given number of rows forwards or backwards in the query. [Atz p141]

Reliability – the DBMS is able to preserve the contents of the DB. [Atz p4] Even if some components or operations fail. [Atz p417]

Renaming – an operator on a relation that changes the name of an attribute. It may be used to allow unions etc of relations with similar but differently-named schemas. [Atz p44]

Replication Rule – an inference rule for MVDs and FDs. $\{X \rightarrow Y\} \models X \twoheadrightarrow Y$. [Elm p517]

Resource – a component of a database (usually a table, view or attribute) on which privileges may be granted or revoked to or from users. [Atz p136]

Restrict – SQL keyword. Used to disallow a drop command if components exist that depend on the “dropped’ component. [Atz p98] SQL keyword. Causes a revoke command to fail if the revocation would cause other privileges to also be revoked. [Atz p138]

ResultSet – a class in java.sql containing the results of the execution of a select command. [Ric p182]

ResultSetMetaData – a class in SQL containing information (attribute names and types) about a ResultSet [Ric p182]

Returns – SQL-3 keyword. Used to denote the type and/or value that a function returns. [Atz p426]

Revoke – SQL keyword. Used to remove privileges from users. Syntax: revoke *priv* on *res* from *users* [restrict | cascade]. [Atz p138]

Right outer join – an outer join r1 $\rhd\lhd_{\text{RIGHT}}$ r2 where dangling tuples from r2 are padded with blanks and inserted into the join. [Atz p53]

Right [Outer] Join – SQL keyword: used to join two tables before selecting from them. See join. [Atz p109]

Role – one end of an association. May have a name, multiplicity and navigability. [Lar p157]

Row type – SQL-3 keyword used to create and manipulate tuple types. [Atz p423]

Scan Operation – a sequential access to all tuples of a table, perhaps performing various operations on the tuples along the way. [Atz p336]

Schema – the characteristics of the data (not the data itself) within the database. [Atz p6]

Schema (of a relation or table) – its heading (or name), followed by (in brackets) the names of its attributes. Eg “TEACHING(Course, Tutor)” [Atz p6]

Schema – SQL keyword specifying that an operation acts on a database schema object. [Atz p90]

Scope for – SQL-3 keyword. Used to specify that the give attribute must take values from a given table. Syntax: ‘scope for *Attr* is *Table*’ [Atz p425]

Scroll – SQL keyword. Used when creating a cursor to indicate that a program should be allowed to move freely across the results of a query. See ‘cursor for’ [Atz p140]

Second – SQL keyword. Used to specify interval attributes. [Atz p90]

Secondary Key – a Candidate Key which was not selected to be the Primary Key. [Elm p485]

Second Normal Form – a nonkey field in a table must be information about the whole key, not just part of the key. [Ken p121] Every nonprime attribute is fully functionally dependent on the primary key. [Elm p488] Alternatively, on some cadidate key (not just the primary key). [Elm p491]

Select – SQL keyword. Used to construct a query. Specifies what attributes to select. [Atz p101] SQL keyword denoting the privilege of being able to query a table, view or attribute via a select statement. [Atz p137]

Selection – an operator that takes a relation and returns a new relation on the same attributes, with only those rows satisfying a given Boolean expression (prepositional formula). [Atz p45,46]

Semantics – the semantics of a schema gives its meaning, that is, how the tables and attributes correspond to real-world things. [Elm p467]

Sequence – Oracle SQL Keyword: 'create sequence *SequenceName*' creates a "sequence" of integers. If a record is inserted with *SequenceName* as the data value, the next interger in the sequence will be inserted. See Identifier Domain. [Bla p278]

Sequentially ordered organisation – a way data may be arranged in memory (primary or secondary). The sequence of tuples is dictated by values assumed by attributes in the tuples. [Atz p324]

Set – SQL keyword. Used with 'update' to modify the data in a table. Specifically, 'set' identifies the attribute(s) to be modified, and their new values. [Atz p130] An unordered collection of objects of the same type, with no duplicates. [Atz p399] SQL-3 keyword. An aggregate operator that returns a set of the grouped data. [Atz p428]

Set constraints – SQL keyword. Used to specify whether a given constraint should be checked every time an operation is performed on the database, or only at the end of a 'transaction'. See immediate, deferred. [Atz p133]

Set default – SQL keyword. Indicates behaviour of the RDBMS when an object is modified (deleted or changed) when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96]

Set null – SQL keyword. Indicates behaviour of the RDBMS when an object is modified (deleted or changed) when there are other objects dependent on it. See cascade, set null, set default, no action, on update, on delete. [Atz p96]

Setof - SQL-3 keyword allowing the construction of sets. [Atz p424]

Set-of – O2 keyword allowing the construction of sets. [Atz p399]

Set-oriented approach – the manipulation of data in sets, as in SQL. See tuple-oriented approach, impedance mismatch. [Atz p139]

Shared Aggregation – An aggregation where the 'part' may belong to more than one 'whole'. Contrast with Composition. [Lar p416]

Show plan – SQL keyword. Used to identify how a particular query will be executed. In particular this will show what indexes (if any) are being used. [Atz p343]

Signature – see 'Method Signature'.

Simple Attribute – basic data type such as text, number, date, time, Boolean, enumerated type, and so on. [Lar p168]

Smallint – SQL keyword. Domain of small integers. See also integer. [Atz p88]

Software Class Hierarchy – a class hierarchy implemented in software. [Lar p409]

Sort Operation – the action of sorting the data stored in the database. [Atz p336]

Specification Class – a class used to contain information about objects of another class. [Lar p141]

Specialization – The process where from a general concept, more specific concepts are derived. See Generalization, Subclass. [Lar p396] A form of migration, where an object moves from a superclass to a subclass. [Atz p410]

Spurious Tuple – a tuple in a join between two (badly designed) relations that contains wrong information, even though the original relations contained correct information. See also 'lossless join property' [Elm p473-475]

SQL – a standard language, the "structured query language", incorporating DDL and DML features, used to manipulate databases. Available in most commercial RDBMSs. [Atz p85]

SQLException – an exception thrown by the executeXXX methods of Statement if there are problems with the statement. [Ric p189]

Sum – SQL keyword. Used to perform an aggregate query that adds up the values of an attribute. [Atz p114]

State (of a database) – see Instance.

State (of an object) – the values of its attributes. [Atz p402]

Statement – a class in java.sql that includes methods for sending SQL commands to the database server. [Ric p182]

Static properties – that part of a class or type definition describing the structure of the objects (that is, the data). See dynamic properties. [Atz p399]

Structural Complexity (of objects) – the fact that objects can be of types formed from arbitrarily nested complex data type constructors. [Atz p400]

Structured Analysis – an analysis of a problem domain that divides it according to functions and procedures. [Lar p132]

Structured Domain – A data type that has a complex structure, eg Address. [Bla p281]

Structured Query Language – See SQL.

Subclass – a specialization of a superclass. Related to the superclass via an 'is-a' relationship. The <*subclass*> is a <*superclass*> [Lar p396]

Superclass – a generalization of a subclass. Related to the subclass via an 'is-a' relationship. The <*subclass*> is a <*superclass*> [Lar p396]

Superficial Equality – Two objects are superficially equal if they have the same state. Superficial equality implies deep equality, but not vice versa. See also identical. [Atz p402]

Superkey – a set of attributes A for a relation r for which there are no two distinct tuples t1 and t2 in r for which t1[A] = t2[A]. (see also key). [Atz p31]

Symbol – words or images used to represent a conceptual class [Lar p131]

Symmetric Association – a Reflexive Association where the two Roles are interchangeable. Example: Person is-a-friend-of Person. See Reflexive Association, Role. [Bla p286]

Synthesis Algorithm – see 'relation synthesis algorithm'.

System – see '_system'.

Table – SQL keyword specifying that an operation acts on a table object. [Atz p91]

Target – the object to which a method is applied. [Atz p404]

Ternary Association – an association between three different classes. [Bla p285]

Theta-join – the selection of a cartesian product. [Atz p54]

Third Normal Form – A nonkey field must be information about the key, not about other nonkey fields. [Ken p121] The relation must satisfy 2NF, and there must be no nonprime attribute which is transitively dependent on the primary key. [Elm p490] Alternatively, if a nontrivial dependency X → A holds, either X is a superkey, or A is a prime attribute. [Elm p491] That is, every nonprime attribute is fully functionally dependent and nontransitively dependent on every key. [Elm p493]

Three-tiered architecture – a system architecture where user applications (databse clients) do not communicate directly with a DBMS (Database server) directly, but rather, communicate with other software (middleware) as an intermediary. [Ric p177]

Time – SQL keyword. Domain of time values. See also timestamp, time zone.

Timestamp – SQL keyword. Domain of date+time values. See also date, time, time zone. [Atz p89]

Time Zone – SQL keyword. See ‘with time zone’. [Atz p89]

To – SQL keyword. Used to specify interval attributes. [Atz p90] Used to specify the user in a ‘grant’ statement. [Atz p137]

Top-down design – a design methodology that begins by creating a conceptual model (a domain model), and translates this into a collection of tables. [Elm p501]

Transformer – a method used to change the data stored in an object. [Atz p405]

Transitive Dependency – a functional dependency $X \rightarrow Y$ is a transitive dependency if there exists a set of attributes Z which is not a subset (or equal to) any key, with $X \rightarrow Z$ and $Z \rightarrow Y$. See ‘third normal form’ [Elm p489]

Transitive Rule – An inference rule for functional dependencies. If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$. [Elm p479] for MVDs: $\{X \twoheadrightarrow Y, Y \twoheadrightarrow Z\} \models X \twoheadrightarrow (Z-Y)$. [Elm p517]

Trivial Functional Dependency – a functional dependency $X \rightarrow Y$ where Y is a subset of X. [Elm p479]

Trivial MVD – a MVD $X \twoheadrightarrow Y$ where either Y is a subset of X or $X \cup Y = R$. [Elm p516]

Tuple – a function from a set of attributes to a collection of elements from the domains of the attributes. Conceptually similar to an ordered pair or triple etc. [Atz p19]

Tuple Constraint – a form of intra-relational integrity constraint which may be evaluated indivisually on single tuples of the relation. [Atz p29,30]

Tuple-oriented approach – the manipulation of sets of data tuple by tuple, as in a typical procedural language. See set-oriented approach, impedance mismatch. [Atz p139]

Tuple Type – in an ORDBMS, a type that describes the data in a row. See also ‘row type’. [Atz p423]

Type – a description of data and method signatures for an object. Sometimes called ‘class’. [Atz p399]

Type Constructor – syntax allowing complex types to be described in terms of simpler ones. [Atz p399]

UML – a notation used for software and conceptual modelling. [Lar p10]

Unavoidable Redundancy – a situation in a database schema where data must be replicated and there is no way to modify the schema and remove the redundancy. [Ken p125]

Under – SQL-3 keyword used to link tables or types in a class hierarchy. [Atz p425]

Unified Modeling Language – see UML.

Uniform Resource Locator – see URL.

Union (of two relations) – the union of two relations r1 and r2 is the set of tuples that belong to either r1 or r2. Note that r1 and r2 must be on the same schema for this to make sense in terms of databases. [Atz p42] SQL keyword. Used to find the 'union' of the rows returned by a 'select' statement (query). [Atz p120]

Union Rule – see Additive Rule.

Unique – SQL keyword. A constraint that the given attribute(s) must take on unique values in the table. Cf. primary key. [Atz p93] SQL keyword. Used to create an index for a key. See 'index'. [Atz p343]

Universal Relation Schema – A (schema for a) single relation containing all the attributes in the database, representing all the information in the database. May be obtained by joining all the tables in the database. [Elm p476, p502]

Universal Relation Assumption – the assumption that every attribute in the universal relation schema has a unique name. [Elm p502]

Unknown Value – a null value indicating that the information exists but is unknown. Eg, the unknown phone number for a person who is known to have a phone. [Atz p23]

Unnesting – flattening out a complex data structure, bringing some data up from one level together with data on higher levels. [Atz p428] Separating nested relations into separate tables so that the schema complies with the first normal form. [Elm p488]

Update – SQL keyword. Used with 'set' to modify data in a table. Specifically, update identifies the table to be modified. [Atz p130] SQL keyword denoting the privilege of being able to modify (update) the data in a table, view or attribute. [Atz p137] SQL keyword indicating that a cursor may be used to modify the data returned in a query. [Atz p140]

Update Anomalies – inconsistencies that arise when the data in a badly designed database is modified. See 'insertion anomalies', 'deletion anomalies', 'modification anomalies'. [Elm p470]

URL – a text string giving the unique location of a resource on a network (typically, on the internet) [Ric p177]

Usage – SQL keyword. Denotes the privilege of being able to use a domain in the definition of a table or schema. [Atz p137]

User – someone who employs the database for his or her own activities. [Atz p9]

User – SQL keyword. In a default specification, indicates that the defaults value is to be the username of the database user. [Atz p92]

Using – SQL keyword. Used in an 'execute' statement to indicate parameters to be fed into the wildcards of a previously prepared SQL command. [Atz p144] Can also be used for the same purpose when opening a cursor. [Atz p145]

Value – the state of an object. [Atz p403]

Value-Based Identity – Objects are distinguished based on their values. The primary key for a table will be a (combination of) natural attribute(s) of the object. [Bla p277]

Value Constraint – see Domain constraint.

Values – SQL keyword. Used with 'insert into' to specify the data values to be added to the table. [Atz p128]

Values for – SQL-3 keyword. See 'are system generated' [Atz p425]

Varchar - SQL keyword. Short for 'character varying' [Atz p88]

Varbit – SQL keyword. Short for 'bit varying'[Atz p88]

Varying – SQL keyword indicating the size of the given attribute may change. [Atz p88]

VBA – Visual Basic for Applications – a programming language used to write event handlers and procedures in Microsoft Access. [Ric p178]

View – usually, a virtual relation. See also materialized view. [Atz p65] SQL keyword: used to manipulate virtual tables. [Atz p133]

Virtual Relation – also called *view*. A derived relation not physically stored in the database, but named and usable in queries as if it were. Contrast with materialized view. [Atz p65]

Visual Basic for Applications – see VBA.

Web server – a software program that waits for requests from a web browser. A typical example of middleware in a three-tiered database architecture. [Ric p177]

Where – SQL keyword. Used to construct queries. Specifies conditions to be satisfied on the attributes returned. [Atz p101] Used in 'delete from' statements, and update… set statements. [Atz p129,130]

With – SQL keyword. Used to add a ‘check option’ to a ‘view’. [Atz p134]

With grant option – SQL keyword. Used to indicate that the user receiving the privilege may also pass it on to others. [Atz p137,138]

With time zone – SQL keyword. Suffixed to time or timestamp domains to indicate the timezone should also be stored.

Year – SQL keyword. Used to help specify interval attributes. [Atz p90]

Notation:

EXAMS(Student, Grade, Course)

- The schema for the table EXAMS, with attributes Student, Grade and Course.

t[Team, Goals] = Liverpool, 3

- The values of a tuple t on the given attributes

$r1 \cup r2$

- Union of two relations: $\{x : x \in r1 \text{ or } x \in r2\}$. Note the relations must be on the same set of attributes

$r1 \cap r2$

- Intersection of two relations (on the same set of attributes). $\{x: x \in r1 \text{ and } x \in r2\}$

$r1 - r2$

- Difference of two relations: $\{x: x \in r1 \text{ and } x \notin r2\}$

$\rho_{A,B \leftarrow C,D}(r1)$

- The "renaming" operator, renaming attribute C of relation r1 to A, and D to B (see [Atz p44-45])

$\sigma_{A<12 \wedge B=\text{"Fish"}}(r1)$

- The "selection" operator, producing a relation on the same attributes as r1, with only those tuples specified by the prepositional (logical) formula in the subscript ([Atz p46-47]).

$\pi_{Name,Age}(r1)$

- The "projection" operator, producing a relation from all the tuples of r1, but using only the attributes specified in the subscript ([Atz p47-48]).

$\neg$

- not

$\wedge$

- and

$\vee$

- or

$\triangleright\triangleleft$

- the "join" operator.

$X \rightarrow Y$

- Y is functionally dependent on X, that is, for any tuples t1 and t2, if t1[X] = t2[X] then t1[Y] = t2[Y].

$F \models X \rightarrow Y$

- The functional dependency $X \rightarrow Y$ can be inferred from the set of functional dependencies F.

F^+

- The closure of a set of functional dependencies F, that is, all functional dependencies that may be inferred from F.
- The closure of a set of functional and multivalued dependencies, that is, all those that may be inferred from F.

X^+

- The closure of a set of attributes X under a set F of functional dependencies. That is, the set of all Y such that $X \rightarrow Y$ is in F^+.

F_{min}

- A minimal cover for F.

$\pi_R(F)$

- The projection of a set of functional dependencies F onto a relation schema R.

$X \twoheadrightarrow Y$

- X multidetermines Y, that is, a multivalued dependency exists.