Definitions

* Abstract superclass (from the UML) – a superclass that has all subclasses shown in its generalization. Contrast with *Concrete superclass*.
* Agile development – an iterative approach for building software, using the software, and then quickly responding to feedback.
* Alternate key – a unique combination of attributes for a table. None of the attributes can be null. A table can have any number of alternate keys. An alternate key is different than a primary key. An alternate key is purely a uniqueness constraint. In contrast, a primary key also provides a preferential means of accessing records. Contrast with *Primary key*.
* Association (from the UML) – a description of a group of relationships among objects with common structure and meaning. Synonym: *Relationship type*. Related: *Link*.
* Association class (from the UML) – an association that is also a class. Like the links of an association, the occurrences of an association class derive identity from the related objects. Like a class, an association class can have attributes, operations, and associations.
* Association end (from the UML) – the intersection of an association with a related class. Synonym: *Role*.
* Attribute – a description of similar values.
* Bus architecture (for a data warehouse) – the practice of building a data warehouse with dimensions that are consistently defined across facts.
* Cardinality – see *Multiplicity*.
* Class (from the UML) – a description of a group of similar objects. Synonym: *Entity type*. Related: *Object*.
* Class model (from the UML) – the UML model that pertains to databases. The class model specifies classes and their relationships.
* Column – the value of an attribute for all records of a database table. There is one value for each record-column intersection.
* Conceptual data model – a data model that focuses on major entity types and relationship types. Provides a high-level overview. Contrast with *Logical data model*, *Physical data model*.
* Concrete superclass (from the UML) – a superclass that has only some subclasses shown in its generalization. Contrast with *Abstract superclass*.
* Connectivity – see *Multiplicity*.
* Constraint – a Boolean condition that data must satisfy in order for it to be stored in a database.
* Data model – a model that describes how data is stored and accessed, usually for a database. Contrast with *Model*.
* Data type – a specification of type and size for the values of an attribute. Examples include long integer, varchar (20), and date. Contrast with *Domain*.
* Data warehouse – a database that is dedicated to data analysis and reporting.
* Database – a permanent, self-descriptive store of data.
* DBMS (database management system) – the software that manages a database.
* Denormalization – a violation of normal forms. Denormalization should only be for good cause, such as to increase performance for a database that is read and seldom updated.
* Derived data – data that can be computed from other data.
* Dimension (for a data warehouse) – a table that specifies a basis for facts.
* Discriminator – an attribute that indicates which subclass further describes each superclass occurrence.
* Domain – a named set of possible values. A domain is a more general concept than a data type. It is a good practice to assign each attribute a domain and then separately resolve the domain to a data type. Domains promote uniform assignment of data types and provide a hook for attaching constraints. Examples include identifier, name, amount, code and description. Contrast with *Data type*.
* Enterprise Architect – a commercial tool for constructing UML models.
* Enterprise data model – a data model that describes an entire organization or some major aspect of an organization.
* Entity (from IE) – see *Object*.
* Entity type (from IE) – see *Class*.
* Entity type, dependent (from IE) – an entity type with identity that depends on other entity types. Its primary key includes one or more foreign keys. Contrast with *Entity type, independent*.
* Entity type, independent (from IE) – an entity type with identity of its own. Its primary key does not include any foreign keys. Contrast with *Entity type, dependent*.
* ERwin – a commercial tool for constructing IE (and IDEF1X) models.
* ETL – acronym for extract, transform, and load. Refers to system scripts for loading a data warehouse.
* Existence-based identity – each entity type has a generated unique number as its primary key. Contrast with *Value-based identity*.
* Fact (for a data warehouse) – a table that measures the performance of a business.
* Field – a synonym for a database column. See *Column*.
* First normal form (of a table) – each row-column combination stores a single value (and not a collection of values).
* Foreign key – a reference to a unique identifier, usually the primary key.
* Generalization (from the UML) – couples a class (the superclass) to one or more variations of the class (subclasses).
* Hillard’s graph complexity – a technique that equates a data model to an undirected graph and computes the complexity of the graph.
* Hoberman’s data quality scorecard**®**. A list of questions for characterizing the quality of a data model drawn from Hoberman’s extensive experience.
* Identity – that property of an object which distinguishes each object from all others.
* IE (Information Engineering) – a prominent data modeling notation that has been in use for many years. Contrast with *UML*.
* Index – an data structure that organizes data into an inverted tree with a wide fan-out. The fan-out is often 50 or more. An index can speed the response to certain queries and enforce uniqueness constraints. An index can be created for one or more columns of a table.
* Key – a generic reference to an alternate key, primary key, or foreign key.
* Link (from the UML) – a relationship among objects. Synonym: *Relationship*. Related: *Association*.
* Logical data model – a data model that fleshes out the conceptual data model with attributes and lesser entity types. Contrast with *Conceptual data model*, *Physical data model*.
* Master data model – a data model that standardizes and cleanses the data for a critical concept. This lets an organization establish a “single version of the truth”.
* Model – an abstraction of some aspect of a problem. Contrast with *Data model*.
* MS-Access – a commercial DBMS product. A non-standard DBMS suitable for prototyping and basic applications.
* Multiplicity (from the UML) – the number of occurrences of one class that may relate to a single occurrence of an associated class. Synonyms: *Cardinality* (though technically, a mathematically incorrect usage), *Connectivity*.
* Natural key – a unique combination of real-world attributes that is used to identify each entity in an entity type.
* Normal form (of a table) – a guideline for relational database design that helps to keep data consistent across table updates.
* Null – a special value denoting that an attribute value is unknown or not applicable.
* Object (from the UML) – a concept, abstraction, or thing that has identity and meaning for an application. Synonym: *Entity*. Related: *Class*.
* OLAP – acronym for on-line analytical processing. Refers to applications involving a data warehouse. Contrast with *OLTP*.
* OLTP – acronym for on-line transaction processing. Refers to operational applications. Contrast with *OLAP*.
* Operation (from the UML) – a function or procedure that can be applied to or by objects in a class.
* Operational data model – a data model that describes a day-to-day application for running the business.
* Ordered association (from the UML) – an association that imposes sequencing on a “many” association end.
* Package (from the UML) – a group of elements (classes, associations, generalizations, and lesser packages) with a common theme. A model consists of one or more packages. Synonym: *Subject area*.
* Pattern – a model fragment that is profound and recurring.
* Physical data model – a data model that converts the logical model into a database design. The emphasis is on physical constructs such as tables, keys, indexes, and constraints. Contrast with *Conceptual data model*, *Logical data model*.
* Primary key – a unique combination of attributes for preferential access to each record in a table. None of the attributes can be null. A table has at most one primary key and normally should have one. IE notation shows the primary key in the top portion of the entity type box. Contrast with *Alternate key*.
* Qualified association (from the UML) – an association in which a qualifier attribute distinguishes among the objects for a “many” association end.
* Qualifier (from the UML) – an attribute that selects among the target objects for a “many” association end, reducing the effective multiplicity, often from “many” to one.
* Record – a collection of values for a database table with one value for each attribute of the table. The values must all pertain to the same identifier. Synonym: *Row*.
* Referential integrity – a database capability that guarantees that a foreign key reference to a primary key really exists.
* Relational database – a database that presents data in the form of tables. Tables have a specific number of columns and an arbitrary number of rows, with a value stored at each row-column intersection.
* Relationship (from IE) – see *Link*.
* Relationship type (from IE) – see *Association*.
* Relationship type, identifying (from IE) -- a relationship type that propagates primary key attributes of the source entity type to the primary key area of the referent entity type. Contrast with *Relationship type, non-identifying*.
* Relationship type, non-identifying (from IE) – a relationship type that propagates primary key attributes of the source entity type to the data area of the referent entity type. Contrast with *Relationship type, identifying*.
* Reverse engineering – the inverse to normal development. Reverse engineers take an existing design and work backwards to infer the underlying logical intent.
* Role (from IE) – see *Association end*.
* Row – see *Record*.
* Schema – the structure of the data in a database.
* Second normal form (of a table) – the table is in first normal form and all non-primary-key attributes depend on the entire primary key.
* Software engineering – a systematic approach to software development that emphasizes thorough understanding prior to design and coding.
* SQL Server – a commercial DBMS product. A robust product suitable for most applications.
* Star schema (for a data warehouse) – The data modeling paradigm of a fact surrounded by multiple dimensions.
* Stored procedure – developer-written code that is stored and executed in the database kernel.
* Subclass (from the UML) – a class that adds specific information for a generalization. Synonym: *Subtype*. Contrast with *Superclass*.
* Subject area (from IE) – see *Package*.
* Superclass (from the UML) – a class that holds common information for a generalization. Synonym: *Supertype*. Contrast with *Subclass*.
* Subtype (from IE) – see *Subclass*.
* Supertype (from IE) – see *Superclass*.
* Table – a database structure with a specific number of columns and an arbitrary number of rows, with a value stored at each row-column intersection.
* Third normal form (of a table) – a table is in second normal form and each non-primary-key attribute directly depends on the primary key.
* TOGAF (The Open Group Architecture Framework) – an approach for designing, planning, implementing, and governing an enterprise information technology architecture.
* Tree -- a set of nodes that connect from child to parent. Each node has one parent node except for the node at the tree’s top. There are no cycles.
* Trigger – database logic that executes upon the occurrence of some event or changing of some condition.
* UML (Unified Modeling Language) – a general purpose software notation. One of the UML’s many models – the class model – is pertinent to databases. Contrast with *IE*.
* Value (from the UML) – a piece of data for an object.
* Value-based identity – a unique combination of real-world attributes is used to identify each entity for an entity type. Contrast with *Existence-based identity*.
* View – a virtual table that is computed on-the-fly from the defining SQL query.
* Zachman Framework – a structure for organizing software development artifacts, created by John Zachman.