**What is RDBMS?**

Short for relational database management system and pronounced as separate letters, a type of database management system (DBMS) that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways.

1. Hierarchical DBMS
2. Network DBMS
3. Object-oriented DBMS
4. Relational DBMS

An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

Objects in a database

1. Database itself
2. Transaction Log
3. Tables
4. Views
5. Synonyms
6. Stored Procedures
7. User Defined Functions
8. Indexes
9. Roles
10. Users

## Life cycle of a relational database

The life cycle of a relational database is the cycle of development and changes that a database goes through during the course of its life. The cycle typically consists of several stages. There is possibility that the database designer/developer can go back to any of the previous stages. This represents an admission that a full understanding of a problem and its solution is likely to evolve as the various stages of design and implementation proceed. The typical eleven stages involved in the life cycle of a [relational database](http://en.wikipedia.org/wiki/Relational_database) are as follows:

1. The designer must try to obtain as complete as **possible an understanding of the real world problem** that is going to be helped by the introduction of a database. This understanding of the nature of the problem and the constraints and outline feasible solutions is often performed using some systems analysis methodology.
2. The **Entity Relationship Diagram (ER Diagram)** is drawn, and this diagram in its modified form serves as an essential part of the logical schema. Attributes of the entity types so produced are then added. [Primary](http://en.wikipedia.org/wiki/Primary_key) and [foreign](http://en.wikipedia.org/wiki/Foreign_key) keys are specified. (Erwin, Visio, Rational Rose, ERStudio, Lucid Chart)
3. [**Normalization**](http://en.wikipedia.org/wiki/Database_normalization) is used to check the [entity-relationship model](http://en.wikipedia.org/wiki/Entity-relationship_model). Some splitting and even recombination of entity types may result from normalization and the entity relationship model will have to be updated accordingly. The entity relationship model and the [table](http://en.wikipedia.org/wiki/Table_%28database%29) definitions resulting from normalization should be consistent.
4. Set of **Table(s) definition** for the required schema is finalized.
5. **The database tables are created. Primary, Foreign keys, database constraints** and [database integrity](http://en.wikipedia.org/wiki/Database_integrity) rules are specified at this stage.
6. At this stage, the file organization is performed. File organization is the way the database relations are to be stored on the storage medium. The file organization is decided on the basis of maximum speed of access, the type of access required and storage space considerations. There are two factors to consider; firstly how the records are to **be physically mapped onto the storage medium**, and secondly **which indexes are to be used** and if so, which fields (attributes, columns) are to be indexed. Indexes are designed to increase the speed of access to required records. **Views can also be defined at this stage**. Views are used to limit access to parts of database only, when used in conjunction with access privileges. Views also make programming simpler.
7. **The designer will be able to design the required** [**queries**](http://en.wikipedia.org/wiki/Query) **at this stage**. The designer should have a good idea of the main types of query and [reports](http://en.wikipedia.org/wiki/Report) the database will have to accommodate.
8. **At this stage, application screens are designed**. The application screens are used to capture the [input](http://en.wikipedia.org/wiki/Input) [information](http://en.wikipedia.org/wiki/Information) that will be kept in the database. Screen design is partially determined by the data items that must be input and output by particular applications and partially in [human-computer interface](http://en.wikipedia.org/wiki/Human-computer_interface) terms. When designing screens, special consideration is given to the suggestions given by the application [end users](http://en.wikipedia.org/wiki/End_user). There are published standards which can be exactly followed for screens design or organization can develop their own screen design standards as per their requirements.
9. **Report design is another area where input from users is paramount**. They will specify what they want to see on the reports and the format of the reports and in the case of regular reports, when they should be produced. Now-a-days most of the application design tools provide easy to use friendly tools for quick reports development. e.g. report builder in [Oracle](http://en.wikipedia.org/wiki/Oracle_database), [Crystal Reports](http://en.wikipedia.org/wiki/Crystal_Reports), [R&R Report Writer](http://en.wikipedia.org/w/index.php?title=R%26R_Report_Writer&action=edit) etc.
10. [**Testing**](http://en.wikipedia.org/wiki/Software_testing) **is performed at this stage**. Application screens, various functions offered by the application screens, data validations through screens and reports are tested and it serves as the ultimate test of the correctness of the [database schema](http://en.wikipedia.org/wiki/Logical_schema) and the viability of the system as a whole. It is recommended to create a test database separate of the production database. The test database will be useful for testing any schema changes and new and modified application before applying the changes to the production (live) database. Careful testing of the system before handover will minimize the expense of later modifications to the schema and major applications.
11. **The final stage is Handover**. This is the stage where the users receive the finished database and applications and begin data entry. In practice, it is likely that the core of the system will be handed over to users and later extensions to the system will be implemented.

**What is SQL ?**

SQL is a database [computer language](http://en.wikipedia.org/wiki/Computer_language) designed for the retrieval and management of [data](http://en.wikipedia.org/wiki/Data) in [relational database management systems](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), database schema creation and modification, and database object access control management.

SQL is a standard interactive and programming language for querying and modifying data and managing databases. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. The core of SQL is formed by a command language that allows you to retrieve, insert, update, and delete data, and perform management and administrative functions. SQL also includes a call-level interface (SQL/CLI) for accessing and managing data and databases remotely.

The first version of SQL was developed at [IBM](http://en.wikipedia.org/wiki/IBM) by [Donald D. Chamberlin](http://en.wikipedia.org/wiki/Donald_D._Chamberlin) and [Raymond F. Boyce](http://en.wikipedia.org/wiki/Raymond_F._Boyce) in the early 1970s. This version, initially called SEQUEL, was designed to manipulate and retrieve data stored in IBM's original relational database product, [System R](http://en.wikipedia.org/wiki/System_R). The SQL language was later formally [standardized](http://en.wikipedia.org/wiki/Standardization) by the [American National Standards Institute](http://en.wikipedia.org/wiki/American_National_Standards_Institute) (ANSI) in 1986. Subsequent versions of the SQL standard have been released as [International Organization for Standardization](http://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO) standards.

Originally designed as a [declarative](http://en.wikipedia.org/wiki/Declarative_programming) query and data manipulation language, variations of SQL have been created by SQL [database management system (DBMS)](http://en.wikipedia.org/wiki/Database_management_system) vendors that add procedural constructs, control-of-flow statements, user-defined data types, and various other language extensions. With the release of the SQL:1999 standard, many such extensions were formally adopted as part of the SQL language via the SQL Persistent Stored Modules (SQL/PSM) portion of the standard.

Common criticisms of SQL include a perceived lack of cross-platform portability between vendors, inappropriate handling of missing data (*see* [*Null (SQL)*](http://en.wikipedia.org/wiki/Null_%28SQL%29)), and unnecessarily complex and occasionally ambiguous language grammar and semantics.

The SQL standard has gone through a number of revisions, as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Name** | **Alias** | **Comments** |
| 1986 | SQL-86 | SQL-87 | First published by ANSI. Ratified by ISO in 1987. |
| [1989](http://en.wikipedia.org/wiki/1989) | SQL-89 | [FIPS](http://en.wikipedia.org/wiki/Federal_Information_Processing_Standard) 127-1 | Minor revision, adopted as FIPS 127-1. |
| [1992](http://en.wikipedia.org/wiki/1992) | [SQL-92](http://en.wikipedia.org/wiki/SQL-92) | SQL2, FIPS 127-2 | Major revision (ISO 9075), Entry Level SQL-92 adopted as FIPS 127-2. |
| [1999](http://en.wikipedia.org/wiki/1999) | [SQL:1999](http://en.wikipedia.org/w/index.php?title=SQL:1999&action=edit) | SQL3 | Added regular expression matching, recursive queries, triggers, support for procedural and control-of-flow statements, non-scalar types, and some object-oriented features. |
| [2003](http://en.wikipedia.org/wiki/2003) | [SQL:2003](http://en.wikipedia.org/wiki/SQL:2003) |  | Introduced [XML](http://en.wikipedia.org/wiki/XML)-related features, window functions, standardized sequences, and columns with auto-generated values (including identity-columns). |
| [2006](http://en.wikipedia.org/wiki/2006) | SQL:2006 |  | ISO/IEC 9075-14:2006 defines ways in which SQL can be used in conjunction with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database and publishing both XML and conventional SQL-data in XML form. In addition, it provides facilities that permit applications to integrate into their SQL code the use of [XQuery](http://en.wikipedia.org/wiki/XQuery), the XML Query Language published by the World Wide Web Consortium ([W3C](http://en.wikipedia.org/wiki/W3C)), to concurrently access ordinary SQL-data and XML documents. |