

**1. WHAT IS DATABASE:**

- A database can be defined as a collection of coherent (match, to according), meaningful data.
- For example the address book is a database and the postal address in the book, is the data that fills the database.

**2. WHAT IS DATABASE MANAGEMENT SYSTEM (DBMS):**

- DBMS is a system that allows inserting, updating, deleting and processing of data.
- To be able to successfully design and maintain database we have to do following:
  - I. Identify which part of the world's data is of interest to us.
  - II. Identify what specific objects in that part of the world's data are of interest.
  - III. Identify a relationship between the object.
- Hence, the objects, their attributes and the relationship between them (that are of interest to us) are stored in the database that is designed, built and populated with data for a specific purpose.
- **Benefits of DBMS:**
  - I. The amount of data redundancy (not needed) in stored data can be reduced.
  - II. No more data inconsistencies.
  - III. Stored data can be shared by a single or multiple users.
  - IV. Data integrity can be maintained. Data integrity refers to the problem of ensuring that database contains only accurate data.
  - V. Security of data can be simply implemented.
  - VI. Data independence can be achieved.

**3. WHAT IS A REALATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):**

- A Relational Database Management System (RDBMS) is a database management system (DBMS) that based on the relational model as introduced by Dr. Edgar F. Codd.
- Relational Database Management System (RDBMS) stores data in the form of related tables. RDMS are powerful because they require few assumptions about how data is related or how it will be extracted from the database. The same database can be viewed in many different ways.
- An important feature of relational systems is that a single database can be spread across several tables. This differs from flat- file database, in which each database is self- contained in a single table.

#### 4. **Dr. E. F. Codd's Rules for RDBMS:**

In 1985, Dr Edgar Codd- A computer scientist working for IBM proposed the relational model for database management which forms the theoretical basis for relational databases.

- **Rule 1: The information Rule:**
  - All the information including metadata (data about data ) has to be represented as stored data in cells of tables”.
  - Also says that the rows and columns have to be strictly unordered.
  
- **Rule 2: Guaranteed Access Rule:**
  - All data should be accessible without ambiguity.
  - This can be accomplished through a **combination of the table name, primary key (unique row), and column name.**
  
- **Rule 3: Systematic Treatment of Null Values:**
  - RDBMS should be capable of allowing each attribute to remain as null, should also support the representation of missing information and inapplicable information”.
  - Has to be handled consistently, cannot be zero or blank.
  - Primary key cannot be Null.
  - Null can be missing data, data that is not applicable or also no data.
  
- **Rule 4: Dynamic On-Line catalog based on the Relational Model:**
  - Database dictionary (catalog) is the structure description of the complete Database and it must be stored online.
  - A relational database must provide access to its structure through the same tools that are used to access the data.
  - This is usually accomplished by storing the structure definition within special system tables.
  
- **Rule 5: Comprehensive Data Sublanguage Rule:**
  - The database must support at least one clearly defined language that includes functionality for
    - **data definition operation** (including view definitions),
    - **data manipulation operation** (update as well as retrieval),
    - security and integrity constraint,
    - **database transaction control** (begin, commit, rollback)..
  - All commercial relational database use forms of standard SQL as their supported comprehensive language.
  -

- **Rule 6: View Updating Rule:**
  - View is nothing but the virtual table, which is temporarily derived from the base tables.
  - All the views that are theoretically updateable must also be updateable by the system.
  
- **Rule 7: High-level Insert, Update, and Delete:**
  - The system must support insert, update, and delete operations at each level of relations.
  - Data can be retrieved from a relational database in set constructed of data from multiple rows and/ or multiple tables.
  - They should also support union, intersection and minus operations.
  
- **Rule 8: Physical Data Independence:**
  - Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.
  - For example, if any file supporting table was renamed or moved from one disk to other then it should not effect the application.
  
- **Rules 9: Logical Data Independence:**
  - Changes made in logical level (table, columns, rows and so on) must not impact and require a change to be made in the application program.
  - How data is viewed should not be changed when the logical structure of the database changes.
  
- **Rules 10: Integrity Independence:**
  - Integrity constraints must be specified separately from application programs and stored in the catalog (database structure). Example : check constraints, trigger etc
  - It must be possible to change such constraints appropriate without unnecessarily affecting existing applications.
  - All databases do preserve two constraints through SQL.
    - No component of primary key can have a null value.
    - If a foreign key is defined in one table, any value in it must exist as a primary in another table.
  
- **Rules 11: Distribution Independence:**

- A user should be totally unaware of whether or not the database is distributed. A variety of reasons make this rule difficult to implement.
- The distribution of portions of the database to various locations should be invisible to users of the database.

➤ **Rules 12: Non subversion Rule:**

- If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change the data.
- This can be achieved by some locking or encryption.
- for example, bypassing a relational security or integrity constraint.

**5. DBMS V/S RDBMS:**

DBMS	RDBMS
In DBMS relationship between two tables or files are maintained programmatically.	In RDBMS relationship between two tables or files can be specified at the time of table creation.
DBMS does not support Client/Server Architecture.	Most of the RDBMS support Client/Server Architecture.
DBMS does not support Distributed database.	Most of the RDBMS supports Distributed databases.
In DBMS there is given an extension in DBMS.	Many tables are grouped in one database in RDBMS.
In DBMS there is no security of data.	In RDBMS there are multiple levels of security. <ol style="list-style-type: none"> <li>1. Logging in at O/S level.</li> <li>2. Command level.</li> <li>3. Object level.</li> </ol>
DBMS may satisfy less than 7 to 8 rules of Dr. E. F. Codd	RDBMS usually satisfy more than 7 to 8 rules of Dr. E. F. Codd
In DBMS using Field, Record, and File.	In RDBMS using Column, Attributes, Row, Tuple, Entity, Table, Relation, Entity class.

**6. FEATURES OF SQL:**

- SQL can be used by a range of users, including those with little or no programming experience.
- It is a non procedural language.

- It reduces the amount of time required for creating and maintain systems.
- It is an English-like language.

### 7. FEATURES OF SQL \*PLUS:

- SQL \*PLUS accepts ad hoc entry of statements.
- It accepts SQL input from files.
- It provides a line editor for modifying SQL statements.
- It controls environment settings.
- It formats query results into basic reports.
- It accesses local and remote database.

### 8. SQL V/S SQL \*PLUS:

SQL	SQL *PLUS
SQL is a language for communicating with the Oracle Server to access data.	SQL *PLUS recognize SQL statement and send them to the server.
SQL is based on American National Standards Institute (ANSI) standard SQL.	SQL *PLUS is the oracle proprietary interface for executing SQL statements.
SQL manipulates data and table definitions in the database.	SQL *PLUS does not allow manipulation of values in the database.
SQL is entered into the AQL buffer on one or more lines	SQL *PLUS is entered one line at a time, not stored in the SQL buffer.
SQL does not have a continuation character.	SQL *PLUS uses a dash (-) as a continuation character if the command is longer than one line.
It cannot be abbreviated.	It can be abbreviated.
SQL uses a termination character to execute commands immediately.	SQL *PLUS does not require termination characters; executes commands immediately.
SQL uses functions to perform some formatting.	SQL *PLUS uses commands to format data.

### 9. COMPONENTS OF SQL:

#### I. DDL(Data Definition Language):

- It is a set of SQL commands used to create, modify and delete database structures but not data. These commands are normally not used by a general user.
- They are normally used by the DBA to a limited extent, a database designer or application developer.
  
- This are the following example of DDL commands:
  - **CREATE**: To create objects in the database.
  - **ALTER**: Alters the structure of the database.
  - **DROP**: Delete objects from the database.
  - **TRUNCATE**: Remove all records from a table, including all space allocated for the records are removed.
  - **COMMENT**: Add comments to the data dictionary.
  - **GRANT**: Gives user's access privileges to database.
  - **RENAME**: Withdraw access privileges given with the GRANT command

## II. **DML (Data Manipulation Language):**

- It is the area of SQL that allows changing data within the database. This are the following example of DML commands:
  - **INSERT**: Insert data into a table.
  - **UPDATE**: Updates existing data within a table.
  - **DELETE**: Deletes all records from a table, the space for the records remain.
  - **CALL**: Call a PL/SQL or Java subprogram.
  - **EXPLAIN PLAN**: Explain access path to data.
  - **LOCK TABLE**: Table control concurrency.

## III. **DCL(Data Control Language):**

- It is the component of SQL statement that control access to data and to the database. This are the following example of DCL commands:
  - **COMMIT**: Save work done.
  - **SAVEPOINT**: Identify a point in transaction to which you can later roll back.
  - **ROLLBACK**: Restore database to original since the last COMMIT.
  - **SET TRANSACTION**: Change transaction options like what rollback segment to use.

- **GRANT / REVOKE:** Grant or take back permissions to or from the oracle users.

IV. **DQL(Data Query Language):**

- It is the component of SQL statement that allows getting data from the database and imposing ordering upon it.
- It includes the SELECT statement. This command is the heart of SQL. It allows getting the data out of the database perform operations with it. Example of DQL command:

- **SELECT:** Retrieve data from the database.