**MODULE 5 : DATABASE**

**BASICS OF DATABASE**

1. **What do you understand By Database**

A database is information that is set up for easy access, management and updating. Computer databases typically store aggregations of [data](https://www.techtarget.com/searchdatamanagement/definition/data) records or files that contain information, such as sales transactions, customer data, financials and product information.

Databases are used for storing, maintaining and accessing any sort of data. They collect information on people, places or things. That information is gathered in one place so that it can be observed and analyzed. Databases can be thought of as an organized collection of information.

**2.What is Normalization?**

**Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

**3.What is Difference between DBMS and RDBMS?**

* DBMS stores the information in files, but RDBMS stores them in tables. Thus, there exists no relationship or link between the data in DBMS.
* DBMS can have repeated data, but RDBMS prevents it by indexing.
* DBMS does not allow normalisation, but RDBMS does.
* DBMS offers low security and speed. In contrast, RDBMS has a higher speed and is more secure.
* DBMSs require less software and hardware. In an RDBMS, hardware and software requirements are comparatively higher.
* DBMS allows only one user, while RDBMS allows multiple users.
* DBMS has less storage capacity.
* Usually, small organisations with fewer data and single users prefer DBMS, but RDBMS is preferable in other situations.
* DBMS handles navigational or hierarchical format, while RDBMS uses primary identifiers and tables for data handling.
* Common examples of DBMS are Windows registry, Microsoft Access, and XML. General examples of RDBMS are SQL, MySQL, Postgres, and Oracle.

**4.What is MF Cod Rule of RDBMS Systems?**

**Rule 1: The Information Rule**

All information, whether it is user information or metadata, that is stored in a database must be entered as a value in a cell of a table. It is said that everything within the database is organized in a table layout.

**Rule 2: The Guaranteed Access Rule**

Each data element is guaranteed to be accessible logically with a combination of the table name, primary key (row value), and attribute name (column value).

**Rule 3: Systematic Treatment of NULL Values**

Every Null value in a database must be given a systematic and uniform treatment.

**Rule 4: Active Online Catalog Rule**

The database catalog, which contains metadata about the database, must be stored and accessed using the same relational database management system.

**Rule 5: The Comprehensive Data Sublanguage Rule**

A crucial component of any efficient database system is its ability to offer an easily understandable data manipulation language ([DML](https://www.geeksforgeeks.org/dml-full-form/)) that facilitates defining, querying, and modifying information within the database.

**Rule 6: The View Updating Rule**

All views that are theoretically updatable must also be updatable by the system.

**Rule 7: High-level Insert, Update, and Delete**

A successful database system must possess the feature of facilitating high-level insertions, updates, and deletions that can grant users the ability to conduct these operations with ease through a single query.

**Rule 8: Physical Data Independence**

Application programs and activities should remain unaffected when changes are made to the physical storage structures or methods.

**Rule 9: Logical Data Independence**

Application programs and activities should remain unaffected when changes are made to the logical structure of the data, such as adding or modifying tables.

**Rule 10: Integrity Independence**

Integrity constraints should be specified separately from application programs and stored in the catalog. They should be automatically enforced by the database system.

**Rule 11: Distribution Independence**

The distribution of data across multiple locations should be invisible to users, and the database system should handle the distribution transparently.

**Rule 12: Non-Subversion Rule**

If the interface of the system is providing access to low-level records, then the interface must not be able to damage the system and bypass security and integrity constraints.

**5.What do you understand By Data Redundancy?**

Data redundancy is a condition created within a database or data storage technology in which the same piece of data is held in two separate places.

This can mean two different fields within a single database, or two different spots in multiple software environments or platforms. Whenever data is repeated, it basically constitutes data redundancy.

Data redundancy can occur by accident but is also done deliberately for backup and recovery purposes.

**6.What is DDL Interpreter?**

DDL is used as an abbreviation for Data Definition Language. DDL refers to a computer language that is primarily used for creating as well as modifying the structure of the database objects present in a database. Such database objects include indexes, tables, schemas, views, and many more.

DDL is also referred to as a data description language in certain contexts since it describes the records and fields in the DB (database) tables.

**7.What is DML Compiler in SQL?**

DML stands for **Data Manipulation Language**. Tables and formulas are helpful when communicating with data stored up to a point in a database through **SQL (Structured Query Language)**, but a time comes when we actually want to execute some fairly complicated data interactions. We will also need the Data Manipulation Language in that situation. DML is a way to inform a database precisely what we want it to do by conversing in a manner that it has been built to comprehend from scratch. When it comes to interacting with existing data, whether adding, moving, or deleting data, it provides a convenient way to do so.

**8.What is SQL Key Constraints writing an Example of SQL Key Constraints**

* + SQL constraints are used to specify rules for the data in a table.
  + There are two SQL Key Constraints:
  + SQL PRIMARY KEY Constraint:
    - The PRIMARY KEY constraint uniquely identifies each record in a table.
    - Primary keys must contain UNIQUE values, and cannot contain NULL values.
    - A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).
    - Example:
    - The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:
    - CREATE TABLE Persons (
    - ID int NOT NULL PRIMARY KEY,
    - LastName varchar(255) NOT NULL,
    - FirstName varchar(255),
    - Age int
    - );
  + SQL FOREIGN KEY Constraint:
    - The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
    - A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
    - The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.
    - Example:
    - The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created:
    - CREATE TABLE Orders (
    - OrderID int NOT NULL PRIMARY KEY,
    - OrderNumber int NOT NULL,
    - PersonID int FOREIGN KEY REFERENCES Persons(PersonID)

**9.What is save Point? How to create a save Point write a Query?**

The SAVEPOINT command in TCL is basically used to temporarily save a transaction so that we can roll back to that point (saved point) whenever required.

Below given is the syntax of the SAVEPOINT TCL command in SQL.

SAVEPOINT savepoint\_name;

After writing our query, we can write "SAVEPOINT;" followed by the savepoint\_name savepoint name, to create a save point and save the transaction to that point.

Also, we can release a savepoint, meaning, remove any particular savepoint. The syntax to release a savepoint is given below:

RELEASE SAVEPOINT savepoint\_name;

**10.What is trigger and how to create a Trigger in SQL?**

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs.

For example, a trigger can be invoked when a row is inserted into a specified table.

As per below syntax Create a Trigger in SQL:

Create trigger [trigger\_name] [before | after]

{insert | update | delete} on [table\_name] [for each row] [trigger\_body]