**MODULE 5**

Q1.) What do you understand by database?

A1.) A database is an organized collection of structured information or data, that is stored in a computer system. They are generally controlled by database management system (DBMS). Together, with data, DBMS and any other applications that helps them function is usually referred to as database system (DBS) or just database.

Q2.) What is Normalization?

A2.) Normalization is a technique that helps reduce data redundancy and eliminates undesirable characteristics like insertion, update, deletion anomalies. Normalization rules divides larger tables into smaller tables using relationships.

Q3.) What is difference between DBMS and RDBMS?

A3.) The differences between DBMS and RDBMS are:

1. In DBMS, data is stored in file format, while in RDBMS, data is stored in table format.
2. There is normalization in DBMS, while in RDBMS, normalization is not achievable.
3. There is no connection between data, while in RDBMS, data in the form of table is linked together.
4. In DBMS, data elements are accessed individually, while in RDBMS, multiple data elements can be accessed together.

Q4.) What is MF Codd’s rule in RDBMS?

A4.) Codd’s rule in DBMS also known as Codd’s 12 rules is a set of thirteen rules that define a database to be a correct RDBMS. The rules are as follows:

1. Rule 0: The Foundation Rule

For a system to be qualified as a RDBMS, it must be able to manage databases entirely through its relational capabilities.

1. Rule 1: The Information Rule

The information in a relational database must be stored in columns or rows or table.

1. Rule 2: The Guaranteed Access Rule:

Each and every data in a relational database must be logically accessible using the combination of the table name, primary key value, column name.

1. Rule 3: Systematic Treatment of NULL Values

NULL values are fully supported in a relational database and represent missing information or inapplicable information in a systematic way, independent of the data type. NULL values are different from empty strings, blank spaces, and 0.

1. Rule 4: Active/Dynamic Online Catalog based on the Relational Model

Database description of a complete database must be stored online. Th rules of the rest of the database must also apply to the catalog.

1. Rule 5: The Comprehensive Data Sublanguage Rule

Relational systems can support multiple languages and different modes of using terminals, such as fill-in-the-blanks mode. However, there must be at least one language whose statements are expressible according to a well-defined syntax.

1. Rule 6: The View Updating Table

Theoretically, updatable views are also practically updatable by the database system.

1. Rule 7: High Level Insert, Update and Delete Rule

The database system must follow high-level relational operations such as insertion, updation, and deletion at each level or row by row. It also supports the union, intersection, and subtraction operations in database systems.

1. Rule 8: Physical Data Independence

The working of a database system should be independent of the physical storage of its data. If a file is modified, it should not interfere with the working of the system.

1. Rule 9: Logical Data Independence

If there’s a change in the logical structure of the database, the user view of the data must not change.

1. Rule 10: Integrity Independence

Integrity constraints specific to a particular relational database must be defined in the relational data sub-language and stored in the catalog and not in the application.

1. Rule 11: Distribution Independence

A database should work properly regardless of its distribution across a network. The end-user should not be able to see that the data is distributed over many locations, they should always get the impressions that the data is located at a single site only.

1. Rule 12: The Non-Subversion Rule

If a relational system allows low-level access, that low cannot be used to subvert or bypass the integrity rules to modify the data. This can be achieved by using some sort of encryption.

Q5.) What do you understand by data redundancy?

A5.) Data redundancy occurs when the same piece of data exists in multiple places, whereas data inconsistency is when the same data exists in different formats in multiple tables. Unfortunately, data redundancy can cause data inconsistency, which can provide a company with unreliable and/or meaningless information.

Q6.) What is DDL Interpreter?

A6.) DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

Q7.) What is DML Compiler in SQL?

A7.) DML Compiler statements in a query language within low level instructions understandable through the query evaluation engine.

Q8.) What is SQL Key Constraints writing an Example of SQL Key Constraints?

A8.) In SQL constraints are predefined rules and restrictions that are enforced in a single column or multiple columns, regarding the values allowed in the columns, to maintain the integrity, accuracy and reliability of that column’s data. There are six main key constraints: Primary, Foreign, SQL NOT NULL, Check and Default.

For example, Primary Key constraint uniquely identifies each record in a table.

CREATE TABLE Persons (

ID int (50) PRIMARY KEY NOT NULL,

LastName varchar (255),

Age int (50)

);

Q9.) What is Save Point? How to create a Save Point write a Query?

A9.) A Save Point is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction. Syntax for Save Point is: SAVEPOINT SAVEPOINT\_NAME. For example,

SAVEPOINT SP1;  
//Savepoint created.  
DELETE FROM Student WHERE AGE = 20;  
//deleted  
SAVEPOINT SP2;  
//Savepoint created.

Q10.) What is Trigger and how to create a Trigger in SQL?

A10.) A Trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. The syntax for trigger is:

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]  [trigger\_body]