

## **MODULE: 5 (DATABASE)**

**Q : 1** What do you understand By Database?

Ans : The database is an organized collection of structured data to make it easily accessible, manageable and update. In simple words a database is a place where the data is stored.

**Q : 2** What is Normalization?

**Ans:** Normalization is the process of organizing the data in the database.

There are three types of normalization in DBMS

1. First normal form
2. Second normal form
3. Third normal form

Definition :

1. First normal form: If a relation contain composite or multi value attribute it violates first normal form.  
A relation is in first normal form if every attributes in that relation its singled valued attribute.
2. Second normal form: Relation is into 2NF if it has no partial dependency is dependent on any proper subset of any candidate key of the table its called second normal form.
3. Third normal form: A relation is 3<sup>rd</sup> normal form if there is no transitive dependency for non prime attribute as well as it is in second normal form.  
A relation is in 3NF if at least one of the following condition holds in every no trivial function dependency.

**Q : 3** What is Difference between DBMS and RDBMS?

**Ans:**

<b>DBMS</b>	<b>RDBMS</b>
DBMS stores data as file.	RDBMS stores data in tabular form.

Data elements need of access individually.	Multiple data elements can be accessed at the same times.
No relationship between data.	Data is stored in the form of tables which are related to each other.
Normalization is not present.	Normalization is present.
DBMS does not support distributed database.	RDBMS supports distributed database.

#### **Q : 4 What is MF Cod Rule of RDBMS Systems?**

**ANS** The Relational model for Database management. Based on relational model. Codd's rule actually define what quality a DBMS requires in order to become a Relational Database Management System(RDBMS). The Codd's 12 rules are as follows.

**Rule 0 – Foundation rule :** Any relational database management system that is propounded to be RDBMS or advocated to be a RDBMS should be able to manage the stored data in its entirety through its relational capabilities.

**Rule 1 – Rule of Information :** Relational Databases should store the data in the form of relations. Tables are relations in Relational Database Management Systems..

**Rule 2 – Rule of Guaranteed Access :** The use of pointers to access data logically is strictly forbidden. Every data entity which is atomic in nature should be accessed logically by using a right combination of the name of table, primary key represented by a specific row value and column name represented by attribute value.

**Rule 3 – Rule of Systematic Null Value Support :** Null values are completely supported in relational databases. They should be uniformly considered as ‘missing information’. Null values are independent of any data type.

**Rule 4 – Rule of Active and online relational Catalog :** In the Database Management Systems lexicon, ‘metadata’ is the data about the database or the data about the data. The active online catalog that stores the metadata is called Data dictionary.

**Rule 5 – Rule of Comprehensive Data Sub-language** : A single robust language should be able to define integrity constraints, views, data manipulations, transactions and authorizations. If the database allows access to the aforementioned ones, it is violating this rule.

**Rule 6 – Rule of Updating Views** : Views should reflect the updates of their respective base tables and vice versa. A view is a logical table which shows restricted data.

**Rule 7 – Rule of Set level insertion, update and deletion** : A single operation should be sufficient to retrieve, insert, update and delete the data.

**Rule 8 – Rule of Physical Data Independence** : Batch and end user operations are logically separated from physical storage and respective access methods.

**Rule 9 – Rule of Logical Data Independence** : Batch and end users can change the database schema without having to recreate it or recreate the applications built upon it.

**Rule 10 – Rule of Integrity Independence** : Integrity constraints should be available and stored as metadata in data dictionary and not in the application programs.

**Rule 11 – Rule of Distribution Independence** : The Data Manipulation Language of the relational system should not be concerned about the physical data storage and no alterations should be required if the physical data is centralized or distributed.

**Rule 12 – Rule of Non Subversion** : Any row should obey the security and integrity constraints imposed. No special privileges are applicable.

**Q : 5** What do you understand By Data Redundancy?

**Ans** 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.

**Q : 6** What is DDL Interpreter?

**Ans** DDL Interpreter DDL expands to Data Definition Language. DDL Interpreter as the name suggests interprets the DDL statements such as schema definition statements like create, delete, etc.

The result of this interpretation is a set of a table that contains the meta-data which is stored in the data dictionary.

**Q : 7** What is DML Compiler in SQL?

**Ans** A DML (data manipulation language) refers to a computer programming language that allows you to insert, delete and (update) data in a database.

A DML is typically a sublanguage of a larger database language like SQL, with the DML containing some of the language's operators.

**Q : 8** What is SQL Key Constraints writing an Example of SQL Key Constraints?

**Ans** SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table.

This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

**PRIMARY KEY :** use to identify data uniquely in a table , primary key is by default not null , never be duplicate , one table can have single primary key. Ex : create table customer(cid int PRIMARY KEY, name varchar(50), city varchar(50), total int);

**UNIQUE KEY :** use to identify data uniquely , it can accept one null value , it can accept data duplication , it can be apply on one or more column of table. Ex : create table customer(cid int , name varchar(50) UNIQUE, city varchar(50), total int);

**FOREIGN KEY:** use to give reference of data from another table , data can be duplicate , this key can apply on one or more table column , it can be nullable.

NOT NULL : value inside this column never be null

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

**Ans** A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax : SAVEPOINT SAVEPOINT\_NAME;

This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

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

**Ans** 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.

Syntax: create trigger [trigger\_name] [before | after] {insert | update | delete} on [table\_name] [for each row] [trigger\_body]