**Module 5:Database**

**Q1. What do you understand by database?**

**Ans.** A database is an organized collection of data that is stored and accessed electronically through the use of a database management system (DBMS), which is a software used to manage data. It refers to related data in a structured form.

**Q2. What is Normalization?**

**Ans.** Database normalization is the process of organizing the attributes of the database to reduce or eliminate **data redundancy (having the same data but at different places)**.

**Problems because of data redundancy:**Data redundancy unnecessarily increases the size of the database as the same data is repeated in many places. Inconsistency problems also arise during insert, delete and update operations.

**Functional Dependency:**Functional Dependency is a constraint between two sets of attributes in relation to a database. A functional dependency is denoted by an arrow (→). If an attribute A functionally determines B, then it is written as A → B.

For example, employee\_id → name means employee\_id functionally determines the name of the employee. As another example in a timetable database, {student\_id, time} → {lecture\_room}, student ID and time determine the lecture room where the student should be.

**Q3. What is difference between dbms and rdbms?**

**Ans.** DBMS stands for **Database Management System** and RDBMS stands for **Relational Database Management System**. The main difference between DBMS and RDBMS is that DBMS stores data as files whereas RDBMS stores data in the form of tables. In DBMS, data is generally stored in either a hierarchical form or a navigational form. In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables. Normalization is not present in DBMS but it is present in RDBMS. DBMS does not apply any security with regards to data manipulation whereas RDBMS defines the integrity constraint for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property. DBMS uses file system to store data, so there will be no relation between the tables

**Q4. What is MF code rule in RDBMS system?**

**Ans.**  Codd's twelve rules are a set of thirteen rules (numbered zero to twelve) proposed by Edgar F. Codd, a pioneer of the relational model for databases, designed to define what is required from a database management system in order for it to be considered relational, i.e., a relational database management system (RDBMS).

**Q5. What do you understand by Data redundancy?**

**Ans.** 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. Many developers consider it acceptable for data to be stored in multiple places. The key is to have a central, master field or space for this data, so that there is a way to update all of the places where data is redundant through one central access point. Otherwise, data redundancy can lead to big problems with data inconsistency, where one update does not automatically update another field. As a result, pieces of data that are supposed to be identical end up having different values.

**Q6.What is DDL Interpreter?**

**Ans.** A DDL interpreter is a component of a database system that processes the data definition language (DDL) statements and records them in tables containing metadata. DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

**Q7.What is DML compiler in SQL?**

**Ans.** DML stand for Dat manipulation Language and it is a group of SQL commands that are used to manipulate data in a database, such as **Select, Insert, Update** and **Delete**. compiler is program that translates DML statement in Query language into low level instruction that the query evaluation engine can understand.

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

**Ans.** SQL key constraints are rules for ensuring the correctness and uniqueness of data in a table. Some common SQL key constraints are:

* NOT NULL: The column value cannot be empty or null.
* UNIQUE: The column value cannot have duplicate values.
* PRIMARY KEY: A combination of NOT NULL and UNIQUE that uniquely identifies each row in a table.
* FOREIGN KEY: A column that references the primary key of another table.

For example, to create a table called students with a primary key constraint on the id column and a not null constraint on the name column, you can use this SQL statement:

CREATE TABLE students

( id INT PRIMARY KEY,

name VARCHAR(50) );

**Q9. What Save point? How to create a Save Point write a Query?**

**Ans.** A save point is a command in SQL that is used to mark a point in a transaction that you can roll back to later if needed. It allows you to undo part of a transaction instead of the whole transaction

To create a save point, you can use the syntax:

SAVEPOINT save point\_name;

where save point\_name is a unique identifier for the savepoint45.

For example, you can create a save point named table\_create before inserting data into a table:

SQL> CREATE TABLE emp\_data ( no NUMBER(3), name VARCHAR(50), code VARCHAR(12) );

Table created.

SQL> SAVEPOINT table\_create;

Save point created.

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

**Ans.** A trigger is a set of SQL statements that automatically execute when specific events occur on a table, such as insert, update or delete12. Triggers help to maintain data integrity, accuracy and consistency1.

To create a trigger, you can use the syntax:

CREATE TRIGGER trigger\_name ON table\_name FOR event\_name AS trigger\_body;

where trigger\_name is a unique identifier for the trigger, table\_name is the name of the table on which the trigger is defined, event\_name is the name of the event that activates the trigger, such as INSERT, UPDATE or DELETE, and trigger\_body is the set of SQL statements that execute when the trigger is fired2345.

For example, you can create a trigger named emp\_audit that inserts a record into an audit table whenever an employee record is updated or deleted:

CREATE TRIGGER emp\_audit ON employee FOR UPDATE, DELETE AS

INSERT INTO audit (emp\_id, action, time) SELECT emp\_id, CASE WHEN DELETED.emp\_id IS NULL THEN ‘INSERT’ ELSE ‘UPDATE’ END, GETDATE() FROM INSERTED UNION ALL SELECT emp\_id, ‘DELETE’, GETDATE() FROM DELETED;