**MODULE: 4 (Database)**

**Q-1.** **What do you understand By Database?**

**A-1.** A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

**Q-2.** **What is Normalization?**

**A-2**.**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.

**Database Normal Forms**

Here is a list of Normal Forms in SQL:

* 1NF (First Normal Form)
* 2NF (Second Normal Form)
* 3NF (Third Normal Form)
* BCNF (Boyce-Codd Normal Form)
* 4NF (Fourth Normal Form)

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

### A-3. Difference between RDBMS and DBMS

|  |  |
| --- | --- |
| **RDBMS** | **DBMS** |
| Data stored is in table format | Data stored is in the file format |
| Multiple data elements are accessible together | Individual access of data elements |
| Data in the form of a table are linked together | No connection between data |
| Normalisation is not achievable | There is normalisation |
| Support distributed database | No support for distributed database |

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

### A-4 Rule 1: Information Rule

A database contains various information, and this information must be stored in each cell of a table in the form of rows and columns.

### Rule 2: Guaranteed Access Rule

Every single or precise data (atomic value) may be accessed logically from a relational database using the combination of primary key value, table name, and column name.

### Rule 3: Systematic Treatment of Null Values

This rule defines the systematic treatment of Null values in database records. The null value has various meanings in the database, like missing the data, no value in a cell, inappropriate information, unknown data and the primary key should not be null.

### Rule 4: Active/Dynamic Online Catalog based on the relational model

It represents the entire logical structure of the descriptive database that must be stored online and is known as a database dictionary. It authorizes users to access the database and implement a similar query language to access the database.

### Rule 5: Comprehensive Data SubLanguage Rule

The relational database supports various languages, and if we want to access the database, the language must be the explicit, linear or well-defined syntax, character strings and supports the comprehensive: data definition, view definition, data manipulation, integrity constraints, and limit transaction management operations. If the database allows access to the data without any language, it is considered a violation of the database.

### Rule 6: View Updating Rule

All views table can be theoretically updated and must be practically updated by the database systems.

### Rule 7: Relational Level Operation (High-Level Insert, Update and delete) Rule

A database system should follow high-level relational operations such as insert, update, and delete in each level or a single row. It also supports union, intersection and minus operation in the database system.

### Rule 8: Physical Data Independence Rule

All stored data in a database or an application must be physically independent to access the database. Each data should not depend on other data or an application. If data is updated or the physical structure of the database is changed, it will not show any effect on external applications that are accessing the data from the database.

### Rule 9: Logical Data Independence Rule

It is similar to physical data independence. It means, if any changes occurred to the logical level (table structures), it should not affect the user's view (application). For example, suppose a table either split into two tables, or two table joins to create a single table, these changes should not be impacted on the user view application.

### Rule 10: Integrity Independence Rule

A database must maintain integrity independence when inserting data into table's cells using the SQL query language. All entered values should not be changed or rely on any external factor or application to maintain integrity. It is also helpful in making the database-independent for each front-end application.

### Rule 11: Distribution Independence Rule

The distribution independence rule represents a database that must work properly, even if it is stored in different locations and used by different end-users. Suppose a user accesses the database through an application; in that case, they should not be aware that another user uses particular data, and the data they always get is only located on one site. The end users can access the database, and these access data should be independent for every user to perform the SQL queries.

### Rule 12: Non Subversion Rule

The non-submersion rule defines RDBMS as a SQL language to store and manipulate the data in the database. If a system has a low-level or separate language other than SQL to access the database system, it should not subvert or bypass integrity to transform data.

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

**A-5.**

* Data redundancy refers to the practice of keeping data in two or more places within a database or data storage system. Data redundancy ensures an organization can provide continued operations or services in the event something happens to its data.
* for example, in the case of data corruption or data loss. The concept applies to areas such as databases, computer memory and file storage systems.
* Data redundancy can occur within an organization intentionally or accidentally. If done intentionally, the same data is kept in different locations with the organization making a conscious effort to protect it and ensure its consistency. This data is often used for backups or disaster recovery.

**Q-6. What is DDL Interpreter?**

**A-6.** DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

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

**A-7.** A data manipulation language (DML) is a computer programming language used for adding (inserting), deleting, and modifying (updating) data in a database.

A DML is often a sublanguage of a broader database language such as SQL, with the DML comprising some of the operators in the language.

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

**A-8.** Constraints are the rules that we can apply on the type of data in a table.

The available constraints in SQL are:

**NOT NULL**: This constraint tells that we cannot store a null value in a column. That is, if a column is specified as NOT NULL then we will not be able to store null in this particular column any more.

**Ex**.

CREATE TABLE Student

(

ID int(6) NOT NULL,

NAME varchar(10) NOT NULL,

ADDRESS varchar(20)

);

**UNIQUE**: This constraint when specified with a column, tells that all the values in the column must be unique. That is, the values in any row of a column must not be repeated.

**Ex**:- CREATE TABLE Student

(

ID int(6) NOT NULL UNIQUE,

NAME varchar(10),

ADDRESS varchar(20)

);

**PRIMARY KEY**: A primary key is a field which can uniquely identify each row in a table. And this constraint is used to specify a field in a table as primary key.

**Ex**:- CREATE TABLE Student

(

ID int(6) NOT NULL UNIQUE,

NAME varchar(10),

ADDRESS varchar(20),

PRIMARY KEY(ID)

);

**FOREIGN KEY**: A Foreign key is a field which can uniquely identify each row in a another table. And this constraint is used to specify a field as Foreign key.

**Ex**:- CREATE TABLE Orders

(

O\_ID int NOT NULL,

ORDER\_NO int NOT NULL,

C\_ID int,

PRIMARY KEY (O\_ID),

FOREIGN KEY (C\_ID) REFERENCES Customers(C\_ID)

)

**CHECK**: This constraint helps to validate the values of a column to meet a particular condition. That is, it helps to ensure that the value stored in a column meets a specific condition.

**Ex**:- CREATE TABLE Student

(

ID int(6) NOT NULL,

NAME varchar(10) NOT NULL,

AGE int NOT NULL CHECK (AGE >= 18)

);

**DEFAULT**: This constraint specifies a default value for the column when no value is specified by the user.

**Ex**:- CREATE TABLE Student

(

ID int(6) NOT NULL,

NAME varchar(10) NOT NULL,

AGE int DEFAULT 18

);

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

**A-9.** Save point command is used to temporarily save a transaction so that you can rollback to that point whenever required.

* Create a Save point:

You can create savepoint within a transaction using the SAVEPOINT statement.

For Example:

SAVEPOINT my\_savepoint;

This statement creates a savepoint named “my\_savepoint” within

The current transaction.

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

**A-10.** A trigger is a stored procedure in a database that automatically invokes whenever a special event in the database occurs.

A trigger is automatically called whenever a data modification event against a table takes place, which is the main distinction between a trigger and a procedure. On the other hand, a stored procedure must be called directly.

Syntax:

*create trigger [trigger\_name]*

*[before | after]*

*{insert | update | delete}*

*on [table\_name]*

*[for each row]*

*[trigger\_body]*