MODULE: 5 (Database)

Basics of Database

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

-> A database is an organized collection of data that can be easily accessed , managed and updated data . and all data store in table form so you can quickly find and use it when it needed.

-> One more example imagine you have library and you want to keep track of all the books.

so data is books title , author , year published , number of copies so you can quickly search for a books or see how many books are avalaible without going through all the book manually.

***2. What is Normalization?***

-> So normalization is process used in databse to organized data in a way that avoid repetition and keeps the data clean and efficient. and break down large , complex tables into smaller , simpler tables without repeating the same data.

-> So normalization helps you keep your database neat , organized and free of unnecessary data duplication .

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

-> DBMS is database management system and RDBMS is relational databse management system all data orgainesed and managed data .

-> In DBMS, data is stored in files or tables, but there’s no strict relationship between different tables.

-> DBMS is suitable for simple databases where complex relationships between data are not required.

-> RDBMS is a more advanced of DBMS that organizes data into tables that are related to each other using keys primary keys and foreign keys.

-> RDBMS is used for more complex databases that require relationships between different types of data.

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

-> Codd’s 12 Rules are a set of thirteen rules proposed by Edgar F. Codd, a pioneer of the relational model for databases

* **The Foundation Rule**:

For any system to qualify as an RDBMS, it must be able to manage databases entirely through its relational capabilities.

* **Rule 1: The Information Rule**:

All information in a relational database is represented explicitly at the logical level and in exactly one way – by values in tables.

* **Rule 2: The Guaranteed Access Rule**:

Each and every datum (atomic value) in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name.

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

Null values must be uniformly treated as “missing information,” not as empty strings, blanks, or zeros.

* **Rule 4: Dynamic Online Catalog Based on the Relational Model**:

The database description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

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

A relational system may support several languages and various modes of terminal use, but there must be at least one language that supports data definition, view definition, data manipulation, integrity constraints, authorization, and transaction boundaries.

* **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**:

The system must support set-level insert, update, and delete operations.

* **Rule 8: Physical Data Independence**:

Changes to the physical storage of data should not affect the application programs.

**Rule 9: Logical Data Independence:**

Changes to the logical structure of the database should not affect the user’s ability to access data

**Rule 10: Integrity Independence**:

Integrity constraints must be specified separately from application programs and stored in the catalog.

**Rule 11: Distribution Independence**:

The distribution of data across multiple locations should be invisible to users.

**Rule 12: Non-Subversion Rule**:

If a system provides a low-level (record-at-a-time) interface, it must not be able to bypass the integrity rules and constraints expressed in the higher-level relational language.

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

* Data Redundancy refers to the situation where the same pieces of data are stored in multiple places within a database or data storage system.
* This can happen intentionally or accidentally.
* Redundancy can be useful for data recovery in case of corruption or loss.
* In computer memory and storage, data redundancy allows for error correction.

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

* A DDL Interpreter is a component within a Database Management System (DBMS) that processes Data Definition Language (DDL) statements.

These statements (like CREATE, ALTER and DELETE) are used to define and modify the structure of database objects like tables, indexes, and schemas

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

The DML Compiler processes Data Manipulation Language (DML)

statements into low-level instructions(machine language)

that can be executed by the database system.

In SQL, DML includes commands like **SELECT**, **INSERT**, **UPDATE**, and **DELETE**, which allow you to manipulate data within tables

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

* NOT NULL Constraint :

Ensures that a column cannot store NULL values.

**CREATE TABLE** Colleges (

college\_id INT ***NOT NULL***,

college\_code VARCHAR(20) ***NOT NULL***,

college\_name VARCHAR(50)

);

* **UNIQUE Constraint** :

Requires that the values in a column must be unique.

That is, the values in any row of a column must not be repeated.

**CREATE TABLE** Colleges (

college\_id INT NOT NULL ***UNIQUE***,

college\_code VARCHAR(20) ***UNIQUE***,

college\_name VARCHAR(50)

);

* PRIMARY KEY Constraint :

Combines NOT NULL and UNIQUE constraints to uniquely identify rows.

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.

**CREATE TABLE** Colleges (

college\_id INT ***PRIMARY KEY***,

college\_code VARCHAR(20) NOT NULL,

college\_name VARCHAR(50)

);

* **FOREIGN KEY Constraint** :

References a record in another table.

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.

**CREATE TABLE** Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

***FOREIGN KEY customer\_id REFERENCES Customers(id)***

);

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

* SAVEPOINT is a mechanism that allows you to create a point within a transaction to which you can later roll back.
* Creating a Savepoint :

To create a savepoint, use the SAVEPOINT statement followed by a name for the savepoint. You can choose any valid identifier as the savepoint name.

For example:

**SAVEPOINT** my\_savepoint;

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

A trigger in SQL is a piece of code that automatically executes in response to specific events occurring on a particular table within the database

* INSERT: When a new row is inserted into the table.
* UPDATE: When an existing row is modified.
* DELETE: When a row is deleted.

***CREATE TRIGGER*** trigger\_name ***[BEFORE | AFTER]*** event **ON** table\_name

**FOR EACH ROW**

BEGIN

-- trigger\_logic

END;