Assignment

Database (Module - 5)

Prepared by:

Nitav Antani

Q1. What do you understand By Database:

Ans:- **database** is an organized collection of data stored and accessed electronically. Small databases can be stored on a file system, while large databases are hosted on computer clusters or cloud storage.

Database management system (**DBMS**) is the software that interacts with end users, applications, and the database itself to capture and analyse the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a **database system**. Often the term "database" is also used loosely to refer to any of the DBMS, the database system or an application associated with the database.

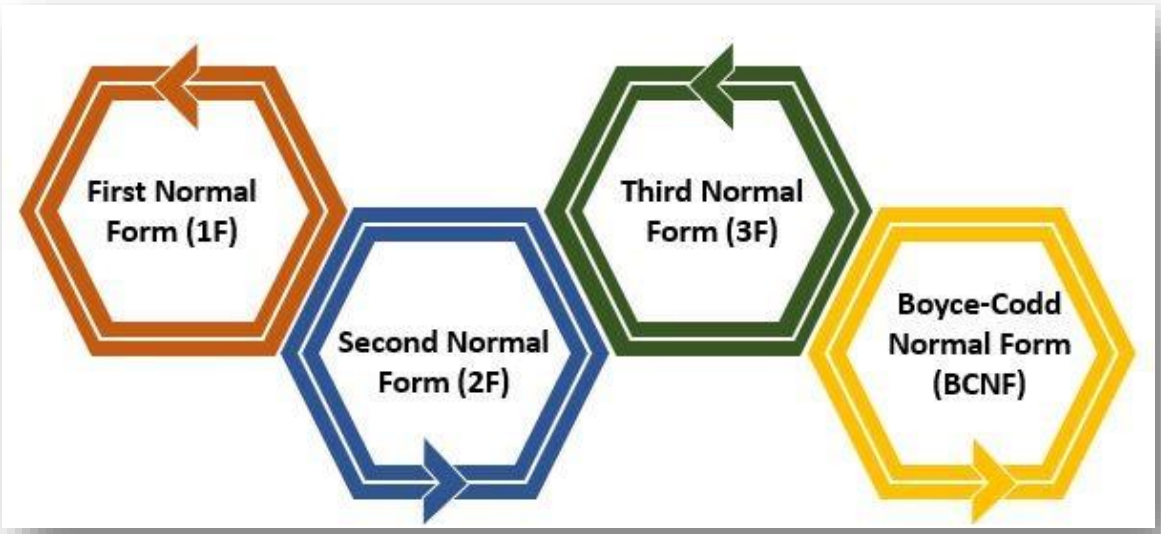
Q2. What is Normalization?

Ans: -

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

There are the four 4 types of database normalization

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



Data redundancy in DBMS means having the same data but at multiple places. It is necessary to remove data redundancy because it causes anomalies in a database which makes it very hard for a database administrator to maintain it.

Q3. What is Difference between DBMS and RDBMS?

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
• Data is stored in a large amount	• Data stored is a small quantity
• Here, redundancy of data is reduced with the help of key and indexes in RDBMS	• Data redundancy is common
• RDBMS supports multiple users	• DBMS supports a single user
• It features multiple layers of security while handling data	• There is only low security while handling data
• The software and hardware requirements are higher	• The software and hardware requirements are low
• Oracle, SQL Server.	• XML, Microsoft Access.

Q4. What is MF Cod Rule of RDBMS Systems?

Ans: -

This rule states that for a system to qualify as an **RDBMS**, it must be able to manage database entirely through the relational capabilities.

These rules were developed by **Dr Edgar F. Codd (E.F. Codd)** in **1985**, who has vast research knowledge on the Relational Model of database Systems.

Rule 1: Information rule

All information (including metadata) is to be represented as stored data in cells of tables. The rows and columns have to be strictly unordered.

Rule 2: Guaranteed Access

Each unique piece of data (atomic value) should be accessible by: **Table Name + Primary Key (Row) + Attribute(column)**.

Rule 3: Systematic treatment of NULL

Null has several meanings, it can mean missing data, not applicable or no value. It should be handled consistently. Also, Primary key must not be null, ever. Expression on NULL must give null.

Rule 4: Active Online CatLog

Database dictionary(CatLog) is the structure description of the complete **Database** and it must be stored online.

Rule 5: Powerful and Well-Structured Language

One well-structured language must be there to provide all manners of access to the data stored in the database. Example: **SQL**, etc.

Rule 6: View Updating Rule

All the view that are theoretically updatable should be updatable by the system as well.

Rule 7: Relational Level Operation

There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

Rule 8: Physical Data Independence

The physical storage of data should not matter to the system. If say, some file supporting table is renamed or moved from one disk to another, it should not affect the application

Rule 9: Logical Data Independence

If there is change in the logical structure (table structures) of the database the user view of data should not change.

Rule 10: Integrity Independence

The database should be able to enforce its own integrity rather than using other programs.

Rule 11: Distribution Independence

A database should work properly regardless of its distribution across a network. Even if a database is geographically distributed, with data stored in pieces, the end user should get an impression that it is stored at the same place. This lays the foundation of **distributed database**.

Rule 12: No subversion Rule

If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change the data. This can be achieved by some sort of looking or encryption.

Q5. 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.

In DBMS, when the same data is stored in different tables, it causes data redundancy.

Let us understand redundancy in DBMS properly with the help of an example.

Student id	Name	Course	Session	Fee	Department
101	Devi	B. Tech	2022	90,000	CS
102	Ruchi	B. Tech	2022	90,000	CS
103	Varun	B. Tech	2022	90,000	CS
104	Satish	B. Tech	2022	90,000	CS
105	Dhwani	B. Tech	2022	90,000	CS

in the above example, there is a "**Student**" table that contains data such as "**student id**", "**Name**", "**Course**", "**Session**", "**Fee**", and "**Departmen**t". As you can see, some data is repeated in the table, which causes redundancy.

Q6. 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.

Data description language (**DDL**) is a syntax for creating and modifying database objects such as tables, indices, and users.

DDL statements are similar to a computer **programming language** for defining **data structures**, especially **database schemas**. Common examples of DDL statements include **CREATE**, **ALTER**, and **DROP**.

Q7. What is DML Compiler in SQL?

Ans: -

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

DML is an abbreviation of **Data Manipulation Language**.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

Attempts to transforms users request within an equivalent and well-organized from for executing the query understandable through Data Manager, Interprets DDL statements and records them within a set of tables containing Meta data in a form that can be used through other elements of a DBMS.

Q8. 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.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Prevents actions that would destroy links between tables
- **CHECK** - Ensures that the values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column if no value is specified
- **CREATE INDEX** - Used to create and retrieve data from the database very quickly

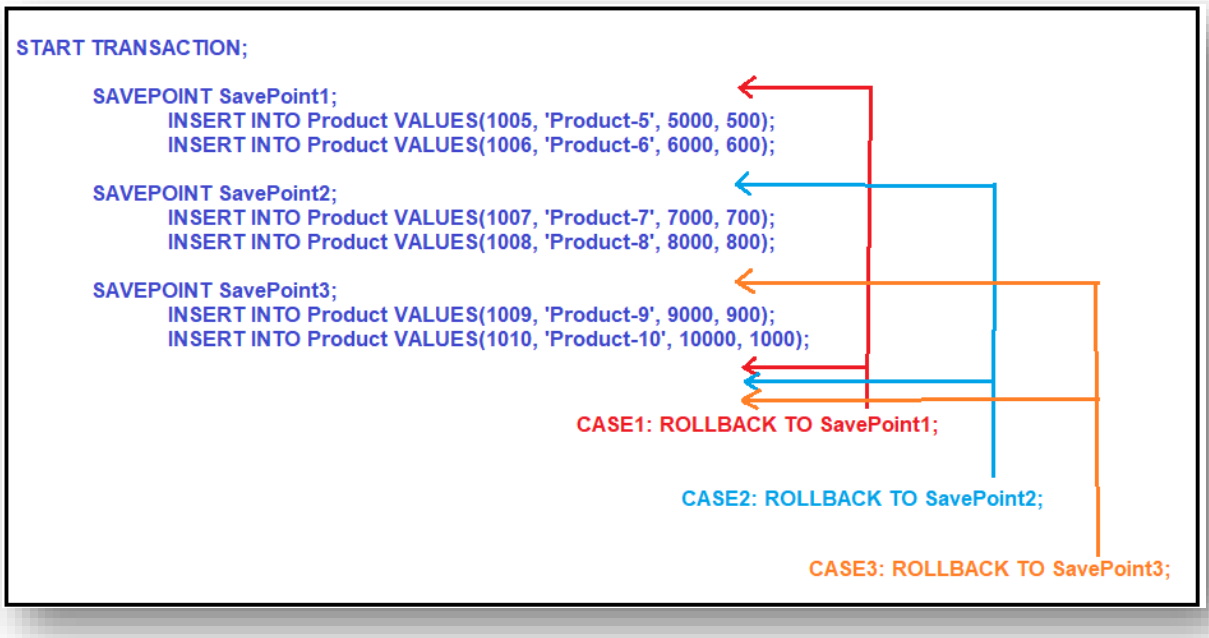
Q9. What is save Point? How to create a save Point write a Query?

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.

the SAVEPOINT statement to create a name for a system change number (SCN), to which you can later roll back.

The SAVEPOINT statement in MySQL is used to save a transaction temporarily. We can save multiple SAVEPOINT in a single transaction.



As per the definition, it will roll back the statements which are present after the SAVEPOINT and before the Rollback Statement. As we created three save points (SavePoint1, SavePoint2, and SavePoint3), let us understand what happens when we execute a particular save point with the rollback command.

CASE1:

ROLLBACK TO SavePoint1;

When we execute the above Rollback command, it will roll back the statements which are starting from SavePoint1, and before the rollback statement. That means in our example, it will roll back all the 6 Insert statements.

CASE2:

ROLLBACK TO SavePoint2;

When we execute the above Rollback statement, then it will roll back the statements which are starting from SavePoint2 and before the Rollback statement. That means, in this case, it will roll back 4 Insert statements.

CASE3:

ROLLBACK TO SavePoint3;

When we execute the above Rollback statement, it will roll back the statements which are present after the SavePoint3 and before the Rollback Command. That means in this case, it will roll back two insert statements.

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

Ans: -

A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.

DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

The CREATE TRIGGER statement allows you to create a new trigger that is fired automatically whenever an event such as **INSERT**, **DELETE**, or **UPDATE** occurs against a table.

The following illustrates the syntax of the CREATE TRIGGER statement:

```
CREATE TRIGGER [schema_name.]trigger_name
ON table_name
AFTER  {[INSERT],[UPDATE],[DELETE]}
[NOT FOR REPLICATION]
AS
{sql_statements}
```

In this syntax:

- The schema_name is the name of the schema to which the new trigger belongs. The schema name is optional.
- The trigger_name is the user-defined name for the new trigger.
- The table_name is the table to which the trigger applies.
- The event is listed in the AFTER clause. The event could be INSERT, UPDATE, or DELETE. A single trigger can fire in response to one or more actions against the table.
- The NOT FOR REPLICATION option instructs SQL Server not to fire the trigger when data modification is made as part of a replication process.
- The sql_statements is one or more Transact-SQL used to carry out actions once an event occurs.