MODULE: 5 (Database)

**Topics Covered Basics of Database**

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

**ANS :**

A database is an organized collection of data, stored and accessed electronically. Databases are used to store and manage large amounts of [structured and unstructured data](https://www.simplilearn.com/structured-vs-unstructured-data-article), and they can be used to support a wide range of activities, including data storage, data analysis, and datamanagement

1. **What is Normalization?**

**ANS :**

Normalization is a **methodological method used in the design of databases** to create a neat, structured, and structured table in which each table relates to just one subject or one-to-one correspondence. The objective is to extensively reduce data redundancy and dependency.

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

**ANS :**

**RDBMS**

* Relation database management system.
* Data Stored is in table format.
* Multiple data element is accessible together.
* Data in the form of a table are linked together.
* Support distributed database.
* Data is Stored in large amount.
* RDBMS supports multiple users.
* The software and hardware requirement are higher.
* Example: - Oracle, SQL, Server.

**DBMS**

* Data stored is in file format
* Individual access of data element
* No connection between data
* No support for distributed database
* Data stored is a small quantity
* DBMS support a single user
* The software and hardware requirements are low
* Example: - XML, Microsoft Assess**.**

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

**ANS :**

[The MF Cod Rule of RDBMS Systems states that **for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities**](https://www.bing.com/ck/a?!&&p=7681d4cee16c9fa1JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc0OQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=4.+What+is+MF+Cod+Rule+of+RDBMS+Systems%3f++&u=a1aHR0cHM6Ly93d3cuc3R1ZHl0b25pZ2h0LmNvbS9kYm1zL2NvZGQtcnVsZS5waHA&ntb=1) . [Rule 0 of the MF Cod Rules states that the system must qualify as relational, as a database, and as a management system. For a system to qualify as an RDBMS, that system must use its relational facilities exclusively to manage the database](https://www.bing.com/ck/a?!&&p=82c96f9cd357cdbaJmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc1Mg&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=4.+What+is+MF+Cod+Rule+of+RDBMS+Systems%3f++&u=a1aHR0cDovL3d3dy5ta2ljcy5pbi9Ob3Rlcy9TWUJDQS9TRU0zL1JEQk1TL0NPREVTLUxBVy1SREJNUy5wZGY&ntb=1).

1. **What do you understand by Data Redundancy?**

**ANS :**

[**Data redundancy refers to the situation where the same pieces of data are stored in multiple places within a database or data storage system**](https://www.bing.com/ck/a?!&&p=55e0be9278185267JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3MQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly93d3cubWluaXRvb2wuY29tL2xpYi9kYXRhLXJlZHVuZGFuY3kuaHRtbA&ntb=1)**.**[**This can happen intentionally or accidentally. Redundancy can be useful for data recovery in case of corruption or loss**](https://www.bing.com/ck/a?!&&p=1713dccd36b64a4bJmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3NQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly9kYXRhY29ub215LmNvbS8yMDIyLzAzLzA5L3doYXQtaXMtZGF0YS1yZWR1bmRhbmN5Lw&ntb=1)**.**[**In computer memory and storage, data redundancy allows for error correction**](https://www.bing.com/ck/a?!&&p=17aadad700ccb733JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTc3Nw&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=5.+What+do+you+understand+By+Data+Redundancy%3f++&u=a1aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRGF0YV9yZWR1bmRhbmN5&ntb=1)

1. **What is DDL Interpreter?**

**ANS :**

DML Compiler: It processes the DML statements into low level instruction (machine language), so that they can be executed. **DDL** Interpreter: It processes the DDL statements into a set of tables containing meta data (data about data)..

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

**ANS :**

The **Data Manipulation Language**, or **DML**for short, is the group of commands responsible for manipulating data in a database; this generally entails inserting, editing, or deleting rows in SQL tables.

* **Query Parser**
* Query Optimizer
* Execution Engine

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

**ANS :**

Constraints are the rules that we can apply on the type of data in a table. That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints.

**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.

* **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.

* **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.

* **FOREIGN KEY**

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

* **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.

* **DEFAULT**

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

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

**ANS :**

[A **save point** in SQL is a **logical rollback point within a transaction**](https://www.bing.com/ck/a?!&&p=a1f26c8649e17103JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxMQ&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly9yZWludGVjaC5pby9ibG9nL2EtZGV0YWlsZWQtZ3VpZGUtb24tc3FsLXNhdmVwb2ludC1zdGF0ZW1lbnQ&ntb=1). [It allows you to specify a point in a transaction that you can roll back to without affecting the entire transaction. To create a, save point, use the following syntax: `SAVEPOINT savepoint\_name`](https://www.bing.com/ck/a?!&&p=f3db28d8b83b0221JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxNA&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly93d3cuc3FsdHV0b3JpYWwubmV0L3NhdmVwb2ludC5odG1s&ntb=1). [You can then perform various SQL operations within the transaction. To roll back to a specific save point, use `ROLLBACK TO save\_point\_name`](https://www.bing.com/ck/a?!&&p=2337a085fcb82b82JmltdHM9MTcxNzk3NzYwMCZpZ3VpZD0zOGNlYzg3OS04MDMyLTY3ODUtM2RjNy1kYjFlODFmMTY2YjcmaW5zaWQ9NTgxNg&ptn=3&ver=2&hsh=3&fclid=38cec879-8032-6785-3dc7-db1e81f166b7&psq=9.+What+is+save+Point%3f+How+to+create+a+save+Point+write+a++Query%3f&u=a1aHR0cHM6Ly9jb2RlYnVyc3QuaW8vc2F2ZXBvaW50LWluLW15c3FsLXdpdGgtZXhhbXBsZXMtMTk3NjkyODk5Mjdh&ntb=1).

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

**ANS :**

A trigger in SQL is a special type of stored procedure that automatically executes in response to certain events on a particular table or view in a database. These events can include INSERT, UPDATE, DELETE operations or a combination thereof. Triggers are used to enforce business rules, validate input data, maintain referential integrity, and automate actions based on database events.

Example:

We are adding tuple to the ‘Donors’ table that is some Person has donated blood. So we can design a trigger that will automatically add the value of donated blood to the ‘blood\_record’ table

We can define 6 types of triggers for each table

* AFTER INSERT: activated after data is inserted into the table.
* AFTER UPDATE: activated after data in the table is modified.
* AFTER DELETE: activated after data is deleted/removed from the table.
* BEFORE INSERT: activated before data is inserted into the table.
* BEFORE UPDATE: activated before data in the table is modified.
* BEFORE DELETE: activated before data is deleted/removed from the table.

**SQL Queries**

|  |  |
| --- | --- |
| **1.** | **Create Table Name: Student and Exam** |
|  | **Create Table Student :**    **Enter Student Data in Table :**    **Student Table:**    **Create Table Exam:**    **Enter Data in Table :**    **Exam Table :** |
| **2** | **Create table given below: Employee and Incentive Table.** |
|  | **Create Table Employee:**    **Enter Data :**    **Employee Table:**    **CREATE TABLE Incentive:**    **Insert Data:**    **Incentive Table:** |
| **3.** | **Get First\_Name from employee table using Tom name “Employee Name”** |
|  |  |
| **4** | **Get FIRST\_NAME, Joining Date, and Salary from employee table.** |
|  |  |
| **5** | **Get all employee details from the employee table order by First\_Name Ascending and Salary descending?** |
|  |  |
| **6** | **Get employee details from employee table whose first name contains ‘J’.** |
|  |  |
| **7** | **Get department wise maximum salary from employee table order by** |
|  |  |
| **8** | **salary ascending?** |
|  |  |
| **9** | **Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000** |
|  |  |
| **10** | **Create After Insert trigger on Employee table which insert records in viewable** |
|  |  |
| **11** | **Create table given below: Salesperson and Customer** |
|  | **Create Table 1: Salesperson**        **Create Table 2: Customer** |
| **12** | **Retrieve the below data from above table** |
| **13** | **All orders for more than $1000.** |
|  |  |
| **14** | **Names and cities of all salespeople in London with commission above 0.12** |
|  |  |
| **15** | **All salespeople either in Barcelona or in London** |
|  |  |
| **16** | **All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).** |
|  |  |
| **17** | **All customers excluding those with rating <= 100 unless they are located in Rome** |
|  |  |
| **18** | **Write a SQL statement that displays all the information about all salespeople** |
|  | **Create Table Salespeople** |
| **19** | **From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.** |
|  | **Create Table Orders**        **Query:** |
| **20** | **From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.** |
|  | **Create Table Item\_mast**        **Query**: |
| **21** | **From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.** |
|  | **Query:** |
| **22** | **From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'** |
|  | **Query:** |
| **23** | **From the following table, write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro\_name and pro\_price.** |
|  | **Query:** |
| **24** | **From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.** |
|  | **Query:** |