**Basics of Database :**

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

ANS :

A database is a way of organizing and storing data in a computer system, so that it can be easily accessed and managed by users or applications. A database usually consists of one or more tables, which contain rows and columns of data. Each row represents a record or an entity, and each column represents a field or an attribute of the entity. For example, a table of students might have columns for name, age, grade, and email.

**2. What is Normalization?**

ANS :

Normalization is a process of organizing data in a database to reduce redundancy and improve consistency. Normalization involves dividing a large table into smaller tables that are linked by relationships, and applying certain rules or constraints to ensure that the data is valid and accurate. Normalization can also help with performance, security, and maintenance of the database.

There are different levels or types of normalization, called normal forms, that indicate how well a table is structured. The most common normal forms are:

**• First normal form (1NF):**

**• Second normal form (2NF):**

**• Third normal form (3NF):**

**• Boyce-Codd normal form (BCNF):**

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

ANS :

A DBMS is a software application that is used to store, manage, and manipulate data in a database. An RDBMS is a type of DBMS that stores data in a relational database, which is a collection of tables that are linked by relationships. The main differences between a DBMS and an RDBMS are:

• A DBMS can store data in various formats, such as files, hierarchies, or networks, but an RDBMS can only store data in tables.

• A DBMS may not enforce any rules or constraints on the data, such as data types, keys, or referential integrity, but an RDBMS must follow these rules to ensure data consistency and accuracy.

• A DBMS may not support advanced features, such as SQL, transactions, views, indexes, or triggers, but an RDBMS must support these features to enable data manipulation and querying.

• A DBMS may not support distributed databases, which are databases that are stored on multiple servers or locations, but an RDBMS must support distributed databases to enable data sharing and scalability.

Some examples of DBMS are file systems, XML, FoxPro, and dBase. Some examples of RDBMS are MySQL, PostgreSQL, SQL Server, Oracle, and Microsoft AccesS

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

ANS :

Codd’s rules are proposed by a computer scientist named Dr. Edgar F. Codd and he also invent the relational model for database management. These rules are made to ensure data integrity, consistency, and usability. This set of rules basically signifies the characteristics and requirements of a relational database management system (RDBMS).

Rule 1: The Information Rule

### Rule 2: The Guaranteed Access Rule

### Rule 3: Systematic Treatment of NULL Values

### Rule 4: Active Online Catalog Rule

### Rule 5: The Comprehensive Data Sublanguage Rule

### Rule 6: The View Updating Rule

### Rule 7: High-level Insert, Update, and Delete

### Rule 8: Physical Data Independence

### Rule 9: Logical Data Independence

### Rule 10: Integrity Independence

### Rule 11: Distribution Independence

### Rule 12: Non-Subversion Rule

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

ANS :

Data redundancy is the situation where the same data is stored more than once in a database or a file system. Data redundancy can be intentional or accidental, and it can have advantages or disadvantages depending on the context. For example, data redundancy can be used to improve data backup, security, and availability, but it can also cause data inconsistency, waste of storage space, and performance issues

**6. What is DDL Interpreter?**

ANS :

A DDL interpreter is a component of a database management system (DBMS) that processes the data definition language (DDL) statements and records them in tables containing metadata. Metadata is data about data, such as the names, types, and constraints of the columns and tables in a database. A DDL interpreter allows users to create, modify, or delete the structure of the database using commands such as CREATE, ALTER, DROP, etc

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

ANS :

A DML compiler is a component of a database management system (DBMS) that processes the data manipulation language (DML) statements and converts them into a low-level instruction that can be understood by the query evaluation engine. The query evaluation engine is the part of the DBMS that executes the instructions and performs the operations on the data, such as selecting, inserting, updating, or deleting

DML statements are the commands that allow users to manipulate the data in a database, such as SELECT, INSERT, UPDATE, and DELETE. A DML compiler analyzes the syntax and semantics of the DML statements and checks for any errors or violations of the database constraints. It also optimizes the DML statements by applying various techniques, such as query rewriting, query decomposition, or query plan generation

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

ANS :

SQL key constraints are rules that ensure the uniqueness and integrity of data in a table. A key is a column or a combination of columns that can identify a row in a table. There are two types of key constraints in SQL: primary key and foreign key

• A primary key constraint defines a column or a set of columns that can uniquely identify each row in a table. A table can have only one primary key, and the primary key columns cannot contain null values.

• A foreign key constraint defines a column or a set of columns that reference a primary key in another table. A foreign key establishes a relationship between two tables and ensures that the values in the referencing table match the values in the referenced table. A table can have multiple foreign keys, and the foreign key columns can contain null values.

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

ANS :

A save point is a way of marking a specific point in a transaction, so that you can roll back to that point if something goes wrong. A save point allows you to undo only a part of a transaction, instead of the whole transaction. You can create multiple save points in a single transaction, and name them as you wish.

To create a save point in SQL, you can use the SAVEPOINT statement, followed by the name of the save point. For example:

SAVEPOINT save\_point\_1;

This creates a save point named save\_point\_1. You can then perform some operations on the data, such as inserting, updating, or deleting rows. If you want to roll back to the save point, you can use the ROLLBACK TO SAVEPOINT statement, followed by the name of the save point. For example:

ROLLBACK TO SAVEPOINT save\_point\_1;

This undoes all the changes made after the save point. You can also release a save point, which means you no longer need it, by using the RELEASE SAVEPOINT statement, followed by the name of the save point. For example:

RELEASE SAVEPOINT save\_point\_1;

This deletes the save point and frees up some memory.

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

ANS :

A trigger is a piece of code that is executed automatically in response to a certain event on a table in a database. A trigger can perform various actions, such as validating data, enforcing business rules, logging changes, or sending notifications.

To create a trigger in SQL, you need to use the CREATE TRIGGER statement. which has the following syntax:

CREATE TRIGGER trigger\_name

[ BEFORE | AFTER ] event

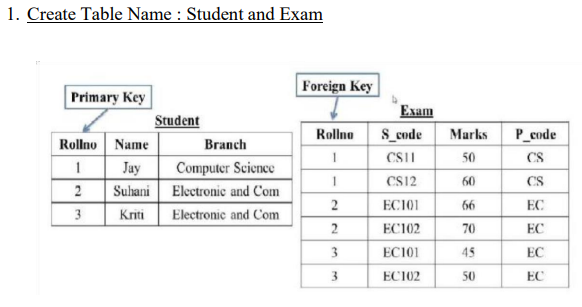
ON table\_name

[ FOR EACH ROW | FOR EACH STATEMENT ]

trigger\_body;

**SQL Queries :**

Q-1. Create Table Name : Student and Exam



ANS- TABLE 1 STUDENT:

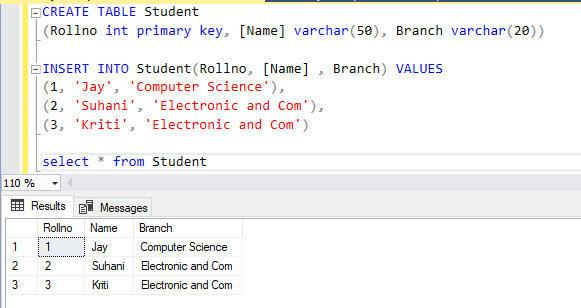
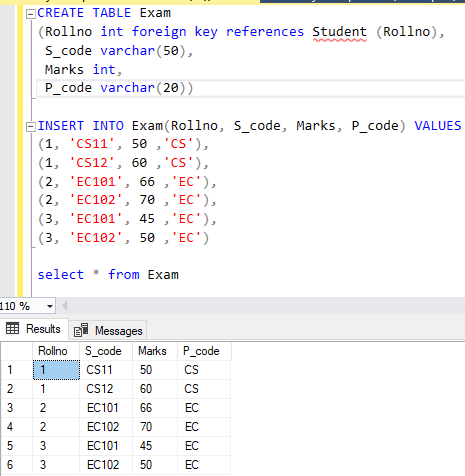
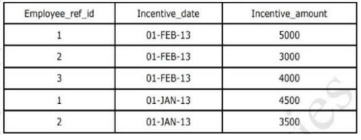


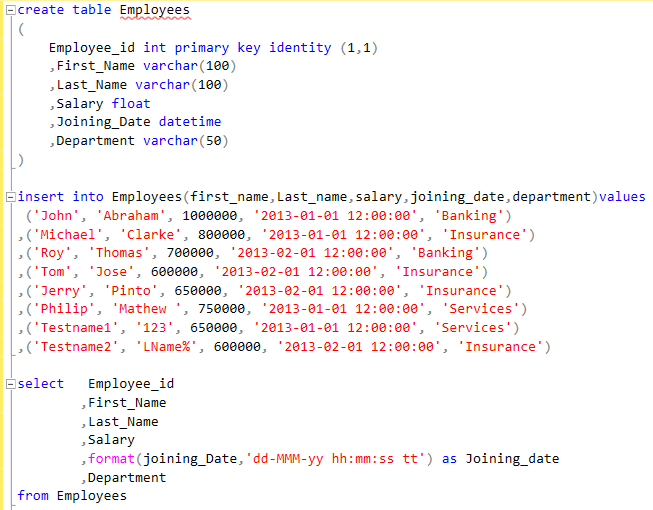
TABLE 2 EXAM:



Q-2. Create table given below: Employee and IncentiveTable



ANS- Table 1 Employee :



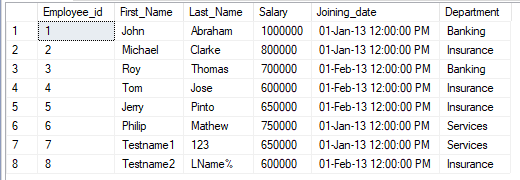
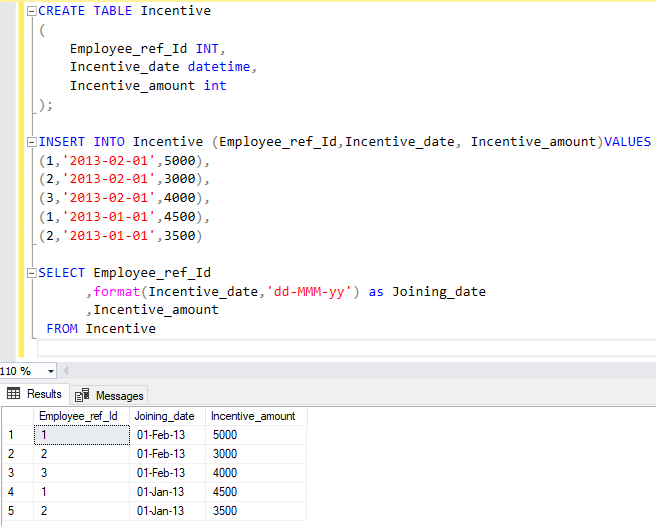
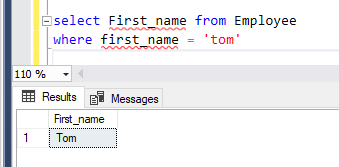


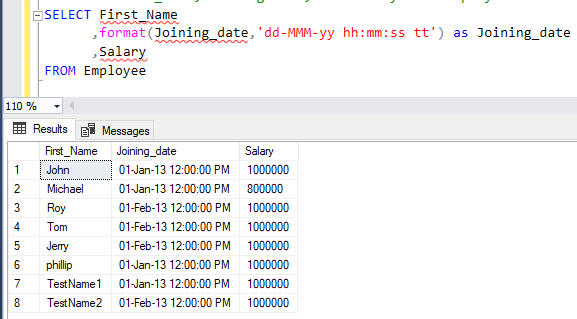
Table 2 INCENTIVE :



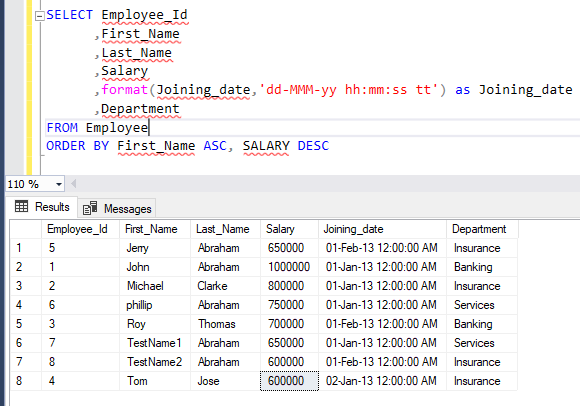
Q-3. Get First\_Name from employee table using Tom name “Employee Name”.



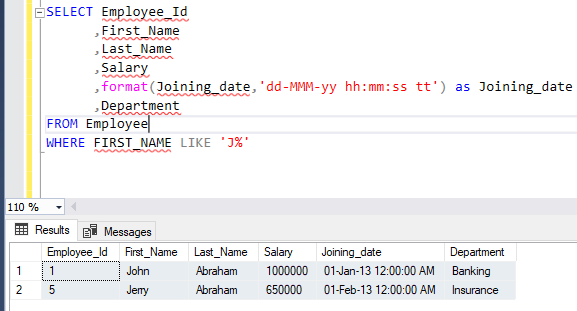
Q-4. Get FIRST\_NAME, Joining Date, and Salary from employee table.



Q-5. Get all employee details from the employee table order by First\_Name Ascending and Salary descending?

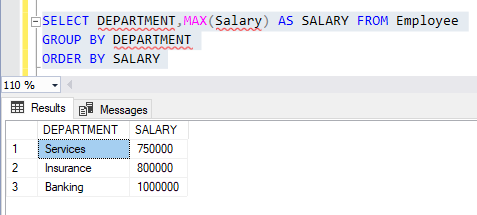


Q-6. Get employee details from employee table whose first name contains ‘J’

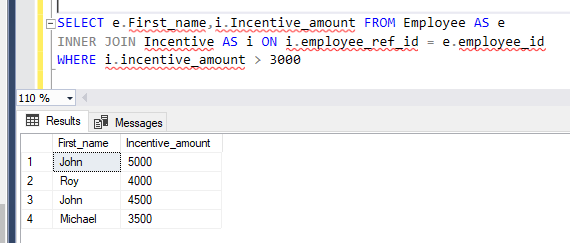


Q-7. Get department wise maximum salary from employee table order by

Q-8. salary ascending?

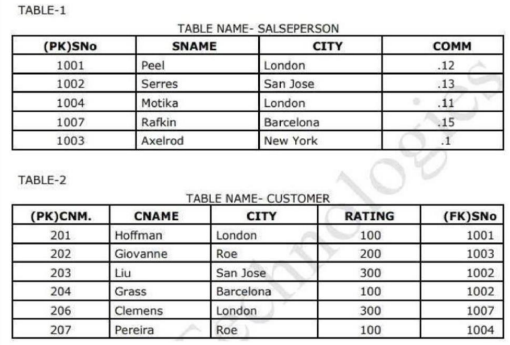


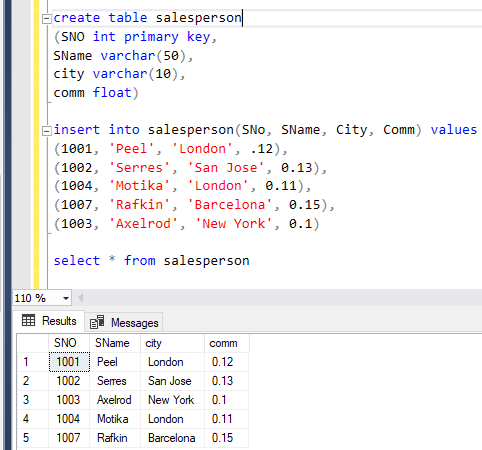
Q-9 Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000

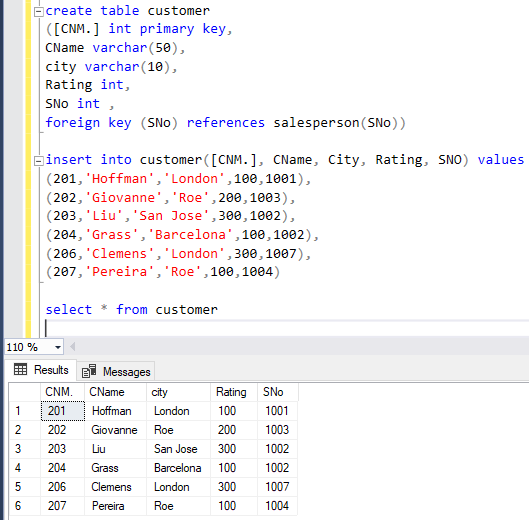


Q-10. Create After Insert trigger on Employee table which insert records in viewtable

Q-11.Create table given below: Salesperson and Customer

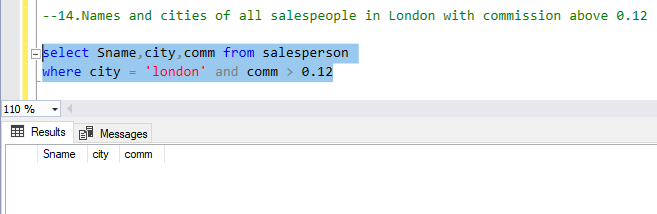




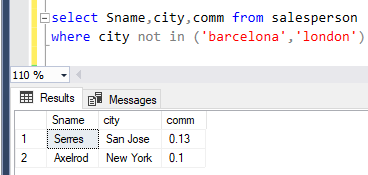


Q-13.All orders for more than $1000.

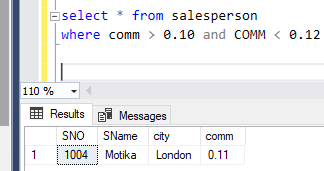
Q-14.Names and cities of all salespeople in London with commission above 0.12



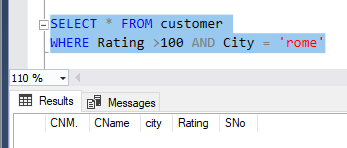
Q-15.All salespeople either in Barcelona or in London



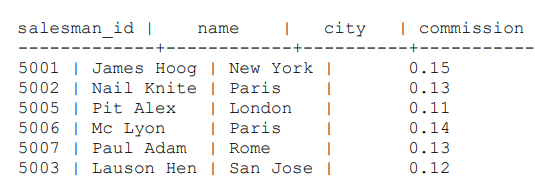
Q-16.All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).



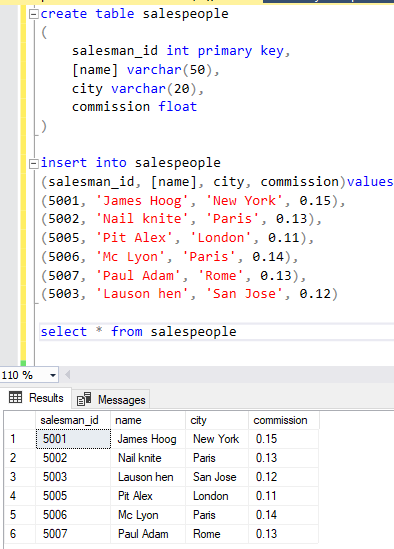
Q-17.All customers excluding those with rating <= 100 unless they are located inRome



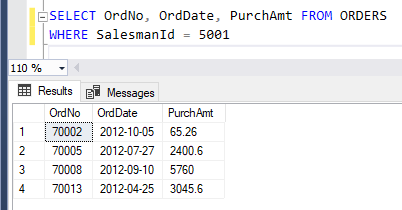
Q-18. Write a SQL statement that displays all the information about all salespeople



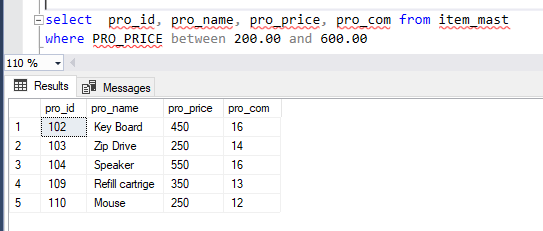
ANS:



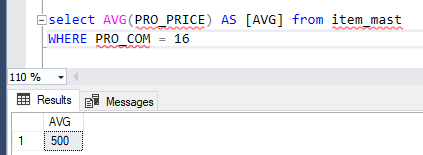
Q-19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.



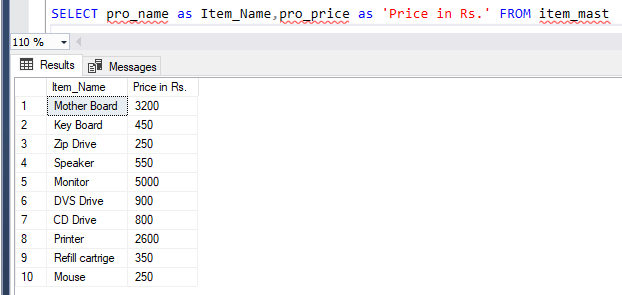
Q-20. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.



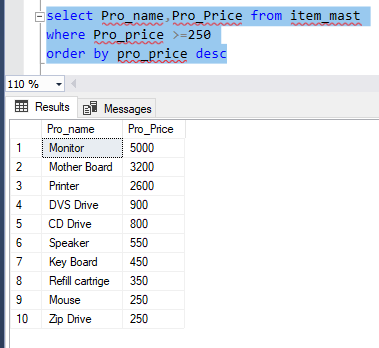
Q-21. From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.



Q-22. From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'



Q-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.



Q-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.

