**MODULE: 5 (Database)**

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

**Ans:** A database is an organized collection of data that is stored and accessed electronically. Databases allow for the efficient storage, retrieval, and management of data. They can be as simple as a file system or as complex as relational databases used in large organizations.

1. **What is Normalization?**

**Ans:** Normalization is the process of organizing data in a database to minimize redundancy and ensure data integrity. It involves dividing large tables into smaller ones and defining relationships between them to eliminate duplication and inconsistencies. There are several forms of normalization, like 1NF (First Normal Form), 2NF (Second Normal Form), and 3NF (Third Normal Form), each progressively reducing redundancy.

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

**Ans:** DBMS (Database Management System): A DBMS is software that helps manage databases by providing an interface to store, retrieve, and manipulate data. DBMS may not always maintain relationships between tables. Examples include file-based systems or hierarchical databases.

RDBMS (Relational Database Management System): RDBMS is a type of DBMS that is specifically designed to handle relational data (tables with rows and columns). It uses SQL (Structured Query Language) for managing the data, and relationships between tables are maintained using foreign keys. Examples include MySQL, PostgreSQL, and Oracle.

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

**Ans:** E.F. Codd's rules (there are 12) define what it means for a system to be considered an RDBMS (Relational Database Management System). Some important rules are:

Rule 0: The system must qualify as relational, which means data is represented in tables.

Rule 1: All information in the system should be represented in the form of tables.

Rule 2: Guaranteed access should be provided to each item through table primary keys. Codd's rules ensure that databases are truly relational, structured, and adhere to strict standards.

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

**Ans:** Data redundancy occurs when the same piece of data is stored in multiple places within a database. This can lead to inefficiency, increased storage costs, and data inconsistencies, as changes in one place might not propagate to others.

1. **What is DDL Interpreter?**

**Ans:** The DDL interpreter is a component in a database system responsible for interpreting Data Definition Language (DDL) statements, which are used to define the database schema (structure). Commands like CREATE, ALTER, and DROP are DDL commands that create or modify database objects like tables, indexes, and constraints.

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

**Ans:** The DML compiler converts Data Manipulation Language (DML) queries (like SELECT, INSERT, UPDATE, and DELETE) into low-level instructions understood by the database's execution engine. DML commands allow users to manipulate the data within a database.

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

**Ans:** SQL key constraints enforce rules at the column level to ensure data integrity. Common key constraints include:

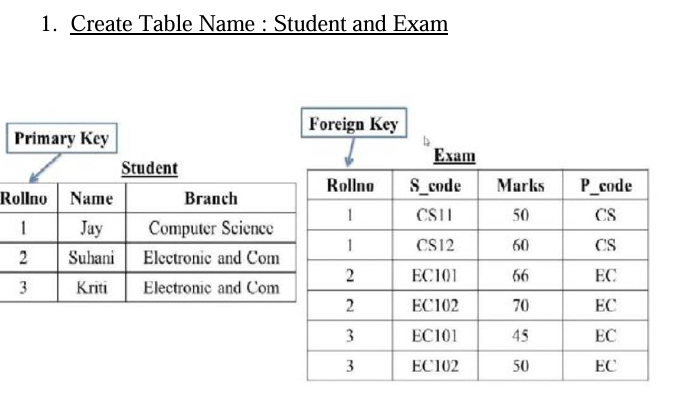
* Primary Key: Ensures that each record in a table is unique and not null.
* Foreign Key: Links a column in one table to the primary key of another table, enforcing referential integrity.
* Unique Key: Ensures all values in a column are unique.

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

**Ans:** A Save Point is a point in a transaction that can be rolled back to without affecting the rest of the transaction. It allows partial rollbacks in case of errors.

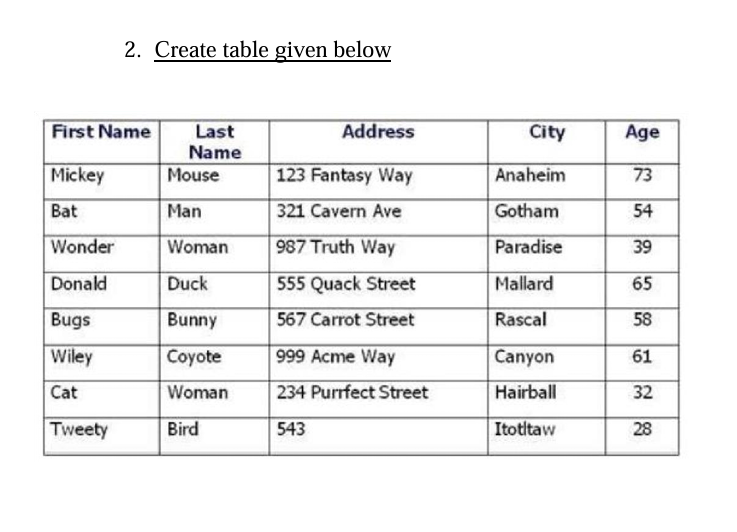
1. **What is trigger and how to create a Trigger in SQL?**

**Ans:** A trigger is a set of SQL commands that automatically execute ("trigger") in response to certain events (like INSERT, UPDATE, or DELETE) on a particular table. Triggers are useful for enforcing business rules, maintaining audit logs, or ensuring consistency.

****

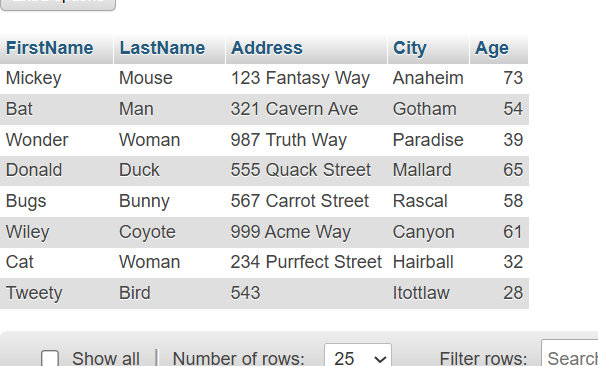
CREATE TABLE Student ( Rollno INT PRIMARY KEY AUTO INCREMENT, Name VARCHAR(50), Branch VARCHAR(50) );

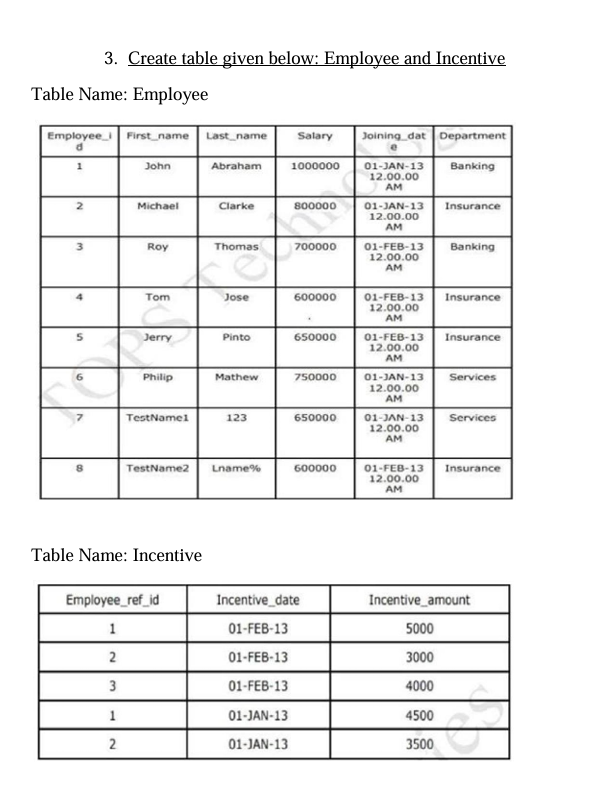
INSERT INTO Student (Rollno, Name, Branch) VALUES (1, 'Jay', 'Computer Science'), (2, 'Suhani', 'Electronic and Communication'), (3, 'Kriti', 'Electronic and Communication');



CREATE TABLE Person ( FirstName VARCHAR(50), LastName VARCHAR(50), Address VARCHAR(100), City VARCHAR(50), Age INT );

INSERT INTO Person (FirstName, LastName, Address, City, Age) VALUES ('Mickey', 'Mouse', '123 Fantasy Way', 'Anaheim', 73), ('Bat', 'Man', '321 Cavern Ave', 'Gotham', 54), ('Wonder', 'Woman', '987 Truth Way', 'Paradise', 39), ('Donald', 'Duck', '555 Quack Street', 'Mallard', 65), ('Bugs', 'Bunny', '567 Carrot Street', 'Rascal', 58), ('Wiley', 'Coyote', '999 Acme Way', 'Canyon', 61), ('Cat', 'Woman', '234 Purrfect Street', 'Hairball', 32), ('Tweety', 'Bird', '543', 'Itottlaw', 28);





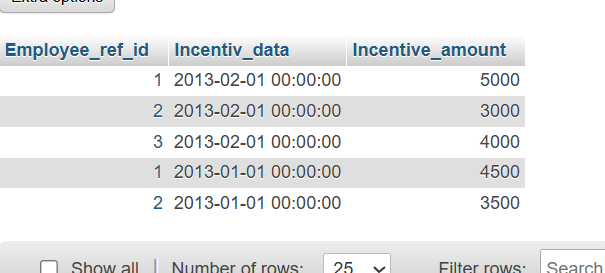
CREATE TABLE Employee ( Employee\_id INT PRIMARY KEY, First\_name VARCHAR(50), Last\_name VARCHAR(50), Salary INT, Joining\_date DATETIME, Department VARCHAR(50) );

INSERT INTO Employee (Employee\_id, First\_name, Last\_name, Salary, Joining\_date, Department) VALUES (1, 'John', 'Abraham', 1000000, '2013-01-01 12:00:00', 'Banking'), (2, 'Michael', 'Clarke', 800000, '2013-01-01 12:00:00', 'Insurance'), (3, 'Roy', 'Thomas', 700000, '2013-02-01 12:00:00', 'Banking'), (4, 'Tom', 'Jose', 600000, '2013-02-01 12:00:00', 'Insurance'), (5, 'Jerry', 'Pinto', 650000, '2013-02-01 12:00:00', 'Insurance'), (6, 'Philip', 'Mathew', 750000, '2013-01-01 12:00:00', 'Services'), (7, 'TestName1', '123', 650000, '2013-01-01 12:00:00', 'Services'), (8, 'TestName2', 'Lname%', 600000, '2013-01-01 12:00:00', 'Insurance');



CREATE TABLE Incentive ( Employee\_ref\_id INT, Incentive\_date DATE, Incentive\_amount INT, FOREIGN KEY (Employee\_ref\_id) REFERENCES Employee(Employee\_id) );

INSERT INTO Incentive (Employee\_ref\_id, Incentive\_date, Incentive\_amount) VALUES (1, '2013-02-01', 5000), (2, '2013-02-01', 3000), (3, '2013-02-01', 4000), (1, '2013-01-01', 4500), (2, '2013-01-01', 3500);



a) Get First\_Name from employee table using Tom name “Employee Name”.

Ans: SELECT First\_name FROM employee1 WHERE Last\_name = 'Tom';

b) Get FIRST\_NAME, Joining Date, and Salary from employee table.

Ans: SELECT First\_name, Joining\_date, Salary FROM employee1;

c) Get all employee details from the employee table order by First\_Name Ascending and Salary descending?

Ans: SELECT \* FROM employee1 ORDER BY First\_Name ASC,Salary DESC;

d) Get employee details from employee table whose first name contains ‘J’.

Ans: SELECT \* FROM employee1 WHERE First\_Name LIKE 'J%';

e) Get department wise maximum salary from employee table order by salary ascending?

Ans: SELECT Department, MAX(Salary)

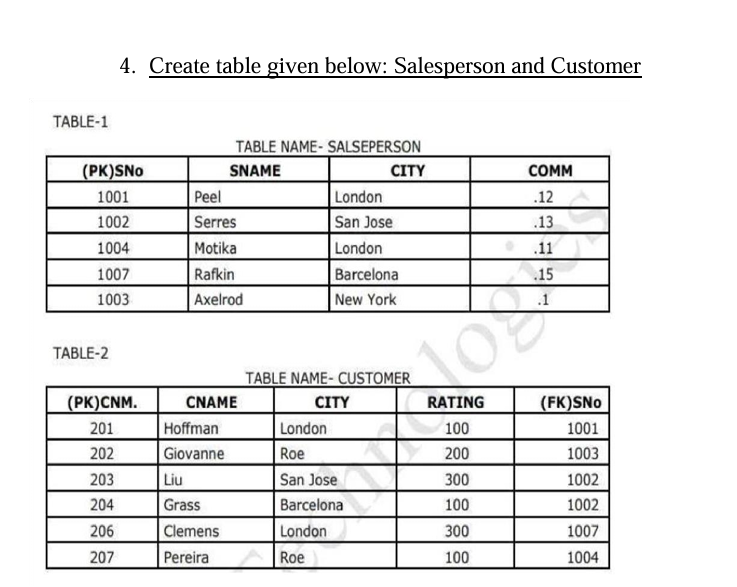
FROM employee1

GROUP BY Department

ORDER BY MAX(Salary) ASC;

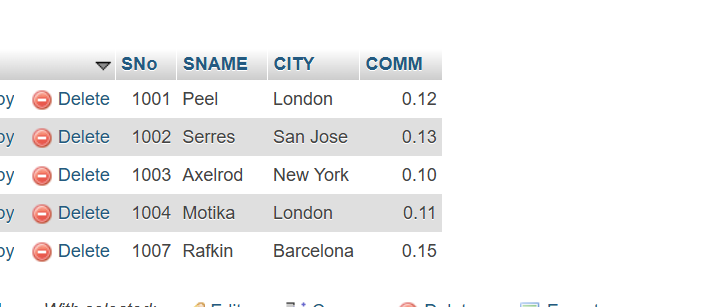
f) Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

Ans: SELECT First\_name, inncentive\_amount FROM employee1 JOIN incentive ON Employee\_id = Employee\_ref\_id WHERE incentive\_amount > 3000;



CREATE TABLE SALESPERSON ( SNo INT PRIMARY KEY, SNAME VARCHAR(50), CITY VARCHAR(50), COMM DECIMAL(5,2) );

INSERT INTO SALESPERSON (SNo, SNAME, CITY, COMM) VALUES (1001, 'Peel', 'London', 0.12), (1002, 'Serres', 'San Jose', 0.13), (1004, 'Motika', 'London', 0.11), (1007, 'Rafkin',  'Barcelona', 0.15), (1003, 'Axelrod', 'New York', 0.1);



CREATE TABLE CUSTOMER ( CNM INT PRIMARY KEY, CNAME VARCHAR(50), CITY VARCHAR(50), RATING INT, SNo INT, FOREIGN KEY (SNo) REFERENCES

SALESPERSON(SNo) );

INSERT INTO CUSTOMER (CNM, CNAME, CITY, RATING, SNo) VALUES

(201, 'Hoffman', 'London', 100, 1001),

(202, 'Giovanne', 'Roe', 200, 1003),

(203, 'Liu', 'San Jose', 300, 1002),

(204, 'Grass', 'Barcelona', 100, 1002),

(206, 'Clemens', 'London', 300, 1007),

(207, 'Pereira', 'Roe',100, 1004);



a) All orders for more than $1000.

Ans: SELECT \* FROM customer WHERE order\_amount > 1000;

b) Names and cities of all salespeople in London with commission above 0.12

Ans: SELECT SNAME, CITY FROM salesperson WHERE CITY = 'London' AND COMM > 0.12;

c) All salespeople either in Barcelona or in London

Ans: SELECT \* FROM salesperson WHERE CITY = 'Barcelona' OR CITY = 'London';

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

Ans: SELECT \* FROM salesperson WHERE COMM > 0.10 AND COMM < 0.12;

e) All customers excluding those with rating <= 100 unless they are located in Rome

Ans: SELECT \* FROM customer WHERE (RATING > 100) OR CITY = 'Rome';