SQL JOINS

NOTE: Create the following dummy tables in MySQL Workbench using CREATE FUNCTION-

TABLE 1: Customers

CustomerID	CustomerName	City
1	John Smith	New York
2	Mary Johnson	Chicago
3	Peter Adams	Los Angeles
4	Nancy Miller	Houston
5	Robert White	Miami

ANSWER- CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

CustomerName VARCHAR(50),

City VARCHAR(50));

INSERT INTO Customers VALUES

(1, 'John Smith', 'New York'),

(2, 'Mary Johnson', 'Chicago'),

(3, 'Peter Adams', 'Los Angeles'),

(4, 'Nancy Miller', 'Houston'),

(5, 'Robert White', 'Miami');

TABLE 2- Orders

OrderID	CustomerID	OrderDate	Amount
101	1	2024-10-01	250
102	2	2024-10-05	300
103	1	2024-10-07	150
104	3	2024-10-10	450
105	6	2024-10-12	400

ANSWER- CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

Amount INT);

INSERT INTO Orders VALUES

(101, 1, '2024-10-01', 250),

(102, 2, '2024-10-05', 300),

(103, 1, '2024-10-07', 150),

(104, 3, '2024-10-10', 450),

(105, 6, '2024-10-12', 400);

TABLE 3: Payments

PaymentID	CustomerID	PaymentDate	Amount
P001	1	2024-10-02	250
P002	2	2024-10-06	300
P003	3	2024-10-11	450
P004	4	2024-10-15	200

ANSWER-CREATE TABLE Payments (

PaymentID VARCHAR(5) PRIMARY KEY,

CustomerID INT,

PaymentDate DATE,

Amount INT);

INSERT INTO Payments VALUES

('P001', 1, '2024-10-02', 250),

('P002', 2, '2024-10-06', 300),

('P003', 3, '2024-10-11', 450),

('P004', 4, '2024-10-15', 200);

TABLE 4: Employees

EmployeeID	EmployeeName	ManagerID
1	Alex Green	NULL
2	Brian Lee	1
3	Carol Ray	1
4	David Kim	2
5	Eva Smith	2

ANSWER-CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

EmployeeName VARCHAR(50),

ManagerID INT);

INSERT INTO Employees VALUES

(1, 'Alex Green', NULL),

(2, 'Brian Lee', 1),

(3, 'Carol Ray', 1),

(4, 'David Kim', 2),

(5, 'Eva Smith', 2);

SELECT * FROM Employees;

QUESTION 1. Retrieve all customers who have placed at least one order.

ANSWER- SELECT DISTINCT c.CustomerID, c.CustomerName

FROM Customers c

INNER JOIN Orders o

ON c.CustomerID = o.CustomerID;

QUESTION 2. Retrieve all customers and their orders, including customers who havenot placed any orders.

ANSWER- SELECT c.CustomerID, c.CustomerName, o.OrderID, o.Amount

FROM Customers c

LEFT JOIN Orders o

ON c.CustomerID = o.CustomerID;

QUESTION 3. Retrieve all orders and their corresponding customers, including orders placed by unknown customers.

ANSWER- SELECT o.OrderID, c.CustomerName, o.Amount

FROM Customers c

RIGHT JOIN Orders o

ON c.CustomerID = o.CustomerID;

QUESTION 4. Display all customers and orders, whether matched or not.

ANSWER- SELECT c.CustomerID, c.CustomerName, o.OrderID, o.Amount

FROM Customers c

LEFT JOIN Orders o

ON c.CustomerID = o.CustomerID

UNION

SELECT c.CustomerID, c.CustomerName, o.OrderID, o.Amount

FROM Customers c

RIGHT JOIN Orders o

ON c.CustomerID = o.CustomerID;

QUESTION 5. Find customers who have **not placed any orders.**

ANSWER- SELECT c.CustomerID, c.CustomerName

FROM Customers c

LEFT JOIN Orders o

ON c.CustomerID = o.CustomerID

WHERE o.OrderID IS NULL;

QUESTION 6. Retrieve customers who made payments but did not place any orders.

ANSWER- SELECT DISTINCT c.CustomerID, c.CustomerName

FROM Customers c

INNER JOIN Payments p

ON c.CustomerID = p.CustomerID

LEFT JOIN Orders o

ON c.CustomerID = o.CustomerID

WHERE o.OrderID IS NULL;

QUESTION 7. Generate a list of all possible combination between Customers and Orders.

ANSWER- SELECT c.CustomerName, o.OrderID

FROM Customers c

CROSS JOIN Orders o;

QUESTION 8. Show all customers along with order and payment amount in one table.

ANSWER- SELECT c.CustomerName,

o.Amount AS OrderAmount,

p.Amount AS PaymentAmount

FROM Customers c

LEFT JOIN Orders o

ON c.CustomerID = o.CustomerID

LEFT JOIN Payments p

ON c.CustomerID = p.CustomerID;

QUESTION 9. Retrieve all customers who have both placed orders and made payments.

ANSWER- SELECT DISTINCT c.CustomerID, c.CustomerName

FROM Customers c

INNER JOIN Orders o

ON c.CustomerID = o.CustomerID

INNER JOIN Payments p

ON c.CustomerID = p.CustomerID;