**DAY-11-Monisha-SQL**

**1)** **Employee–Department Data Analysis Using Joins**

**1)CREATE TABLES:**

create table departments (

department\_id int primary key,

department\_name varchar(100)

);

create table labours (

employee\_id int primary key,

employee\_name varchar(100),

department\_id int,

salary int,

foreign key (department\_id) references departments(department\_id)

);



**2)INSERT DATA:**

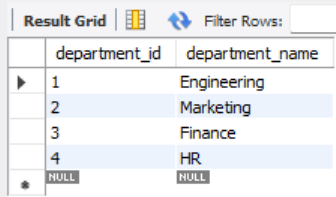
INSERT INTO departments (department\_id, department\_name) VALUES

(1, 'Engineering'),

(2, 'Marketing'),

(3, 'Finance'),

(4, 'HR');



SELECT \* FROM departments;

INSERT INTO labours (employee\_id, employee\_name, department\_id, salary) VALUES

(101, 'Senthil', 1, 70000),

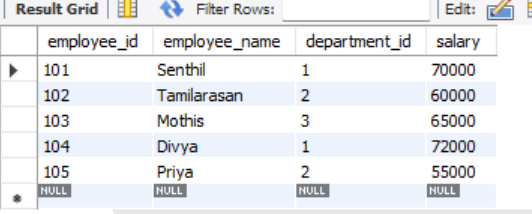
(102, 'Tamilarasan', 2, 60000),

(103, 'Mothis', 3, 65000),

(104, 'Divya', 1, 72000),

(105, 'Priya', 2, 55000);

SELECT \* FROM labours;



**3)INNER JOIN:**

SELECT

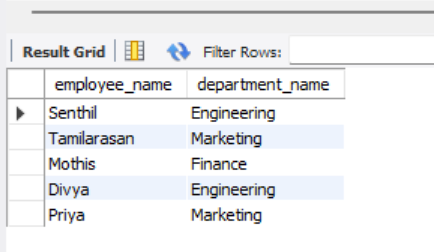
e.employee\_name,

d.department\_name

FROM labours e

INNER JOIN departments d

ON e.department\_id = d.department\_id;



**4)** **LEFT JOIN**:

SELECT

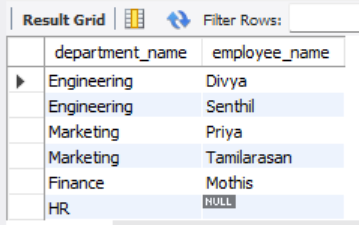
d.department\_name,

e.employee\_name

FROM departments d

LEFT JOIN labours e

ON d.department\_id = e.department\_id;



**5)JOIN WITH AGGREGATION:**

SELECT

d.department\_name,

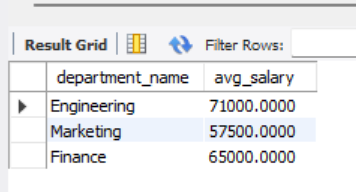
AVG(e.salary) AS avg\_salary

FROM labours e

JOIN departments d

ON e.department\_id = d.department\_id

GROUP BY d.department\_name;



**2)Customer Segmentation Using Subqueries**

**1)CREATE TABLES:**

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE

);

CREATE TABLE ratings (

rating\_id INT PRIMARY KEY,

customer\_id INT,

product\_id INT,

rating INT

);

****

**2)INSERT INTO DATA:**

INSERT INTO orders (order\_id, customer\_id, order\_date) VALUES

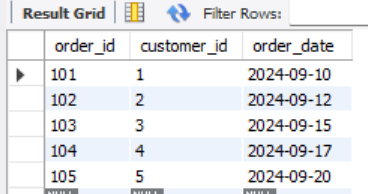
(101, 1, '2024-09-10'),

(102, 2, '2024-09-12'),

(103, 3, '2024-09-15'),

(104, 4, '2024-09-17'),

(105, 5, '2024-09-20');



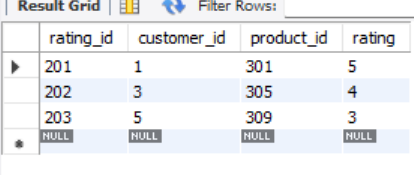
SELECT \* FROM orders;

INSERT INTO ratings (rating\_id, customer\_id, product\_id, rating) VALUES

(201, 1, 301, 5),

(202, 3, 305, 4),

(203, 5, 309, 3);

****

SELECT \* FROM ratings;

**3)USING A SUBQUERY**

SELECT customer\_id

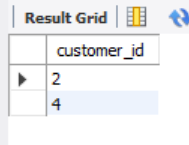
FROM orders

WHERE customer\_id NOT IN (

SELECT customer\_id

FROM ratings

);



--ALTERNATIVE WITH A JOIN:we can solve using this LEFT JOIN

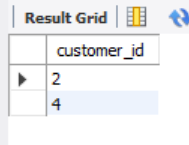
SELECT o.customer\_id

FROM orders o

LEFT JOIN ratings r

ON o.customer\_id = r.customer\_id

WHERE r.customer\_id IS NULL;



**3)PRODUCT-SALES ANALYSIS**

**1)CREATE TABLE:**

CREATE TABLE products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

category VARCHAR(100)

);

CREATE TABLE sales (

sale\_id INT PRIMARY KEY,

product\_id INT,

quantity INT,

price\_per\_unit DECIMAL(10,2),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

**2)INSERT INTO DATA:**

INSERT INTO products (product\_id, product\_name, category) VALUES

(1, 'Laptop', 'Electronics'),

(2, 'Headphones', 'Electronics'),

(3, 'T-shirt', 'Clothing'),

(4, 'Jeans', 'Clothing'),

(5, 'Blender', 'Home Appliances');

INSERT INTO sales (sale\_id, product\_id, quantity, price\_per\_unit) VALUES

(101, 1, 5, 70000.00),

(102, 2, 10, 3000.00),

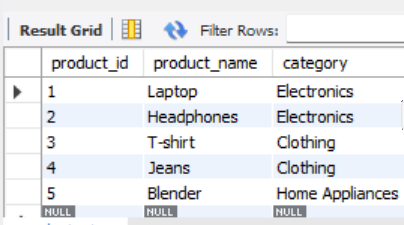
(103, 3, 20, 800.00),

(104, 4, 15, 1500.00),

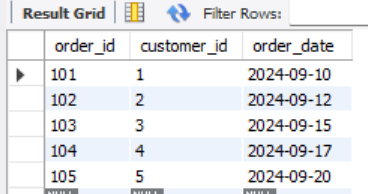
(105, 5, 8, 4000.00);

**View Tables:**

SELECT \* FROM products;

****

SELECT \* FROM salee;

****

**3)Find Total Sales Revenue per Category:**

SELECT

p.category,

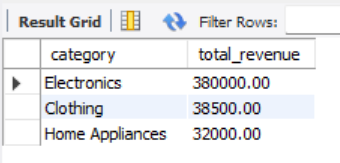
SUM(s.quantity \* s.price\_per\_unit) AS total\_revenue

FROM products p

JOIN salee s

ON p.product\_id = s.product\_id

GROUP BY p.category;



**4) Employee Performance with Subqueries**

**1)CREATE TABLE:**

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(100),

department VARCHAR(100),

performance\_score DECIMAL(5,2)

);

**2)INSERT DATA:**

INSERT INTO employes (id, name, department, performance\_score) VALUES

(1, 'Alice', 'Engineering', 85.5),

(2, 'Bob', 'Marketing', 78.0),

(3, 'Charlie', 'Finance', 90.2),

(4, 'David', 'Engineering', 88.0),

(5, 'Emma', 'HR', 70.5);



**3)FIND THE EMPLOYEES ABOVE AVERAGE:**

SELECT

name,

department,

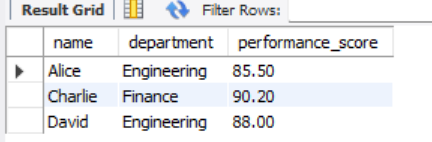
performance\_score

FROM employes

WHERE performance\_score > (

SELECT AVG(performance\_score) FROM employes

);

****

**4)INNER QUERY**:

SELECT AVG(performance\_score) FROM employes;

