**Day 10- Assignments**

**---------------------------------------------------------------------------------------------------------------------**

**Assignment 1**: Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.

**Solution**:

**Create customers Table**

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(100) NOT NULL,

region VARCHAR(50)

);

**Create orders Table**

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

order\_date DATE,

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

* **INNER JOIN Query**

This query combines the orders and customers tables to retrieve details of customers from a specified region who have placed orders.

SELECT customers.customer\_id, customers.customer\_name, orders.order\_id, orders.order\_date

FROM customers

INNER JOIN orders ON customers.customer\_id = orders.customer\_id

WHERE customers.region = 'specified\_region';

Replace 'specified\_region' with the actual region you are interested in.

* **LEFT JOIN Query**

This query displays all customers, including those who have not placed any orders.

SELECT customers.customer\_id, customers.customer\_name, orders.order\_id, orders.order\_date

FROM customers

LEFT JOIN orders ON customers.customer\_id = orders.customer\_id;

**Assignment 2**: Utilize a sub query to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

**Solution**:

* **Create customers table**

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255) NOT NULL

);

* **Create orders table**

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_value DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

* **Insert Sample Data**

-- Insert data into customers table

INSERT INTO customers (customer\_id, customer\_name) VALUES

(1, 'John Doe'),

(2, 'Jane Smith'),

(3, 'Alice Johnson'),

(4, 'Bob Brown');

-- Insert data into orders table

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

(1, 1, 100.00),

(2, 1, 150.00),

(3, 2, 200.00),

(4, 2, 50.00),

(5, 3, 300.00),

(6, 3, 100.00),

(7, 3, 150.00),

(8, 4, 20.00),

(9, 4, 30.00),

(10, 4, 25.00),

(11, 4, 35.00),

(12, 4, 40.00);

* **Subquery to Find Customers with Orders Above Average Order Value**

SELECT c.customer\_id, c.customer\_name

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_value> (

SELECT AVG(order\_value)

FROM orders

);

* **UNION Query to Combine Two SELECT Statements**
* Customers with Orders Above Average Value

SELECT c.customer\_id, c.customer\_name

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_value> (

SELECT AVG(order\_value)

FROM orders

)

UNION

* Customers with More Than 5 Orders

SELECT c.customer\_id, c.customer\_name

FROM customers

JOIN orders o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id, c.customer\_name

HAVING COUNT(o.order\_id) > 5;