

```

-- =====
- Lab Experiment 05 - SQL Joins --
=====
Objective: -- To understand the concept of SQL joins and their applications in relational
databases. -- To implement different types of joins: Inner Join, Outer Join (Left, Right, Full), and
Natural Join. --
=====
Step 1: Create Database DROP DATABASE IF EXISTS Lab05_Joins; CREATE DATABASE
Lab05_Joins; -- output -- 23:10:06 CREATE DATABASE Lab05_Joins 1 row(s) affected 0.000
sec USE Lab05_Joins; -- output -- 23:10:54 USE Lab05_Joins 0 row(s) affected 0.000 sec --
Step 2: Create Tables CREATE TABLE Customers ( customer_id INT PRIMARY KEY,
customer_name VARCHAR(50), city VARCHAR(50) ); -- output -- 23:11:22 CREATE TABLE
Customers ( customer_id INT PRIMARY KEY, customer_name VARCHAR(50), city
VARCHAR(50) ) 0 row(s) affected 0.032 sec CREATE TABLE Orders ( order_id INT PRIMARY
KEY, customer_id INT, product_name VARCHAR(50), order_date DATE, FOREIGN KEY
(customer_id) REFERENCES Customers(customer_id) ); -- output -- 23:11:54 CREATE TABLE
Orders ( order_id INT PRIMARY KEY, customer_id INT, product_name VARCHAR(50),
order_date DATE, FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ) 0
row(s) affected 0.047 sec -- Step 3: Insert Sample Data INSERT INTO Customers (customer_id,
customer_name, city) VALUES (1, 'A', 'Bangalore'), (2, 'B', 'Delhi'), (3, 'C', 'Mumbai'), (4, 'D',
'Chennai'); -- output -- 23:12:47 INSERT INTO Customers (customer_id, customer_name, city)
VALUES (1, 'A', 'Bangalore'), (2, 'B', 'Delhi'), (3, 'C', 'Mumbai'), (4, 'D', 'Chennai') 4 row(s)
affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec INSERT INTO Orders (order_id,
customer_id, product_name, order_date) VALUES (101, 1, 'Laptop', '2023-12-15'), (102, 2,
'Smartphone', '2024-01-10'), (103, 1, 'Headphones', '2024-02-20'), (104, 3, 'Tablet', '2024-03-
05'); -- output -- 23:13:19 INSERT INTO Orders (order_id, customer_id, product_name,
order_date) VALUES (101, 1, 'Laptop', '2023-12-15'), (102, 2, 'Smartphone', '2024-01-10'), (103,
1, 'Headphones', '2024-02-20'), (104, 3, 'Tablet', '2024-03-05') 4 row(s) affected Records: 4
Duplicates: 0 Warnings: 0 0.016 sec --
=====
LAB TASKS --
=====
1. INNER JOIN -- Purpose: Returns rows that have matching values in both tables. -- Task: Find
all orders placed by customers from the city "Bangalore". -- WRITE YOUR QUERY BELOW --
2. LEFT OUTER JOIN -- Purpose: Returns all rows from the left table, even if there are no
matches in the right table. -- Task: Find all customers and their orders, even if a customer has
no orders. -- WRITE YOUR QUERY BELOW --
3. RIGHT OUTER JOIN -- Purpose: Returns all rows from the right table, even if there are no matches in the left table. -- Task: Find all orders
and their corresponding customers, even if an order doesn't have a customer associated with it.
-- WRITE YOUR QUERY BELOW --
4. FULL OUTER JOIN -- Purpose: Returns all rows when
there is a match in either left or right table. -- Task: Find all customers and their orders, including
those customers with no orders and orders without a customer. -- NOTE: MySQL does not
directly support FULL OUTER JOIN. -- WRITE YOUR QUERY BELOW (Hint: Use UNION of
LEFT JOIN and RIGHT JOIN) --
5. NATURAL JOIN -- Purpose: Similar to an inner join, but
automatically joins on columns with the same name in both tables. -- Task: Assume Customers
and Orders have common column customer_id. Find all orders placed by customers. -- WRITE
YOUR QUERY BELOW

```