KONERU LAKSHMAIAH EDUCATION FOUNDATION
(Deemed to be University estd, u/s, 3 of the UGC Act, 1956)
(NAAC Accredited "A++" Grade University)
Green Fields, Guntur District, A.P., India – 522502
Department of Computer Science and Engineering

Active Learning Method

Program: B. Tech Academic Year / Yr-Sem : 2024 - 25 / II - II Sem

Course Title & Code: **DBMS & 23AD2102R**

Date: Time: Venue:

CO2	3
Topics	Advanced SQL Concepts: Joins
Type of ALM	Case Study
Learning Approach	Participatory Learning

Activity: Write sql queries using various types of joins in an e-commerce database scenario.

Task: Write SQL queries that demonstrate the use of different types of joins (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN, CROSS JOIN, and SELF JOIN) based on the following requirements.

Questions:

1. Retrieve a list of all orders along with the first and last names of the customers who placed them.

```
SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

2. List all customers and their corresponding orders, including customers who have not placed any orders.

```
SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName,
Orders.OrderID, Orders.OrderDate
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

3. List all orders and their associated customers, including orders where the customer information might be missing.

```
SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerID,
Customers.FirstName, Customers.LastName
FROM Orders
RIGHT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

4. List all customers and all orders, including customers without orders and orders without customer information.

```
SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName,
Orders.OrderID, Orders.OrderDate
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

5. Generate a list of all possible combinations of customers and products available in the store.

```
SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName,
Products.ProductID, Products.ProductName
FROM Customers
CROSS JOIN Products;
```

6. If there is an Employees table where each employee has a ManagerID referencing the EmployeeID within the same table, list all employees along with their managers.

```
SELECT E.EmployeeID, E.FirstName, E.LastName, M.EmployeeID AS ManagerID,
M.FirstName AS ManagerFirstName, M.LastName AS ManagerLastName
FROM Employees E
LEFT JOIN Employees M ON E.ManagerID = M.EmployeeID;
```