**SQL JOINS**

**In SQL, a JOIN clause is used to combine rows from two or more tables based on a related column between them. There are several types of joins that serve different purposes:**

**INNER JOIN: Returns records that have matching values in both tables.**

**LEFT JOIN (or LEFT OUTER JOIN): Returns all records from the left table and the matched records from the right table. If there is no match, the result is NULL on the right side.**

**RIGHT JOIN (or RIGHT OUTER JOIN): Returns all records from the right table and the matched records from the left table. If there is no match, the result is NULL on the left side.**

**FULL JOIN (or FULL OUTER JOIN): Returns all records when there is a match in either left or right table. If there is no match, the result is NULL from the side where there is no match.**

**CROSS JOIN: Returns the Cartesian product of both tables, which means it returns all possible combinations of rows from the tables.**

**SELF JOIN: A regular join but the table is joined with itself.**

### Sample Database Schema

**Customers Table:**

| **CustomerID** | **CustomerName** |
| --- | --- |
| **1** | **John Doe** |
| **2** | **Jane Smith** |
| **3** | **Michael Brown** |
| **4** | **Emily Davis** |

**Orders Table:**

| **OrderID** | **CustomerID** | **OrderDate** | **ProductID** |
| --- | --- | --- | --- |
| **101** | **1** | **2023-01-10** | **1** |
| **102** | **3** | **2023-02-15** | **2** |
| **103** | **4** | **2023-03-05** | **3** |
| **104** | **1** | **2023-04-12** | **2** |

**Products Table:**

| **ProductID** | **ProductName** |
| --- | --- |
| **1** | **Laptop** |
| **2** | **Smartphone** |
| **3** | **Tablet** |
| **4** | **Headphones** |

### Query Questions

1. **INNER JOIN: List all orders along with the customer name and product name.**

**SELECT Orders.OrderID, Customers.CustomerName, Products.ProductName, Orders.OrderDate**

**FROM Orders**

**INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID**

**INNER JOIN Products ON Orders.ProductID = Products.ProductID;**

1. **LEFT JOIN: List all customers and their orders, if any. Include customers with no orders.**

**SELECT Customers.CustomerID, Customers.CustomerName, Orders.OrderID, Orders.OrderDate**

**FROM Customers**

**LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;**

1. **RIGHT JOIN: List all orders and the respective customer names. Include orders with no matching customers.**

**SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate**

**FROM Orders**

**RIGHT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;**

1. **FULL JOIN: List all customers and orders, showing customers with no orders and orders with no customers.**

**SELECT Customers.CustomerID, Customers.CustomerName, Orders.OrderID, Orders.OrderDate**

**FROM Customers**

**FULL JOIN Orders ON Customers.CustomerID = Orders.CustomerID;**

1. **SELF JOIN: List all customers who have the same name as another customer.**

**SELECT A.CustomerID AS CustomerID1, A.CustomerName, B.CustomerID AS CustomerID2**

**FROM Customers A**

**INNER JOIN Customers B ON A.CustomerName = B.CustomerName AND A.CustomerID <> B.CustomerID;**

1. **COMPLEX JOIN: List all customers and their orders, along with product names. Include customers with no orders and orders with no customers.**

**SELECT Customers.CustomerID, Customers.CustomerName, Orders.OrderID, Orders.OrderDate, Products.ProductName**

**FROM Customers**

**FULL JOIN Orders ON Customers.CustomerID = Orders.CustomerID**

**LEFT JOIN Products ON Orders.ProductID = Products.ProductID;**

1. **NATURAL JOIN: List all orders along with the product names using a natural join.**

**SELECT Orders.OrderID, Orders.OrderDate, Products.ProductName**

**FROM Orders**

**NATURAL JOIN Products;**

1. **CROSS JOIN: Generate a list of all possible combinations of customers and products.**

**SELECT Customers.CustomerName, Products.ProductName**

**FROM Customers**

**CROSS JOIN Products;**