Practical 5 Part II

Sushmit Partakke
23070521156
B2

What is a Join?

A **JOIN** combines records from two or more tables using a related column.

Types of Joins:

1. **INNER JOIN** → Returns only matching records.
2. **LEFT JOIN** → Returns all records from the left table and matching records from the right table.
3. **RIGHT JOIN** → Returns all records from the right table and matching records from the left table.
4. **FULL OUTER JOIN** → Returns all records from both tables (not available in MySQL).
5. **CROSS JOIN** → Returns the Cartesian product of both tables.
6. **SELF JOIN** → Joins a table to itself.

1. Customer Table

Column	Data Type	Constraints
customer_id	NUMBER (PK)	PRIMARY KEY, AUTO-INCREMENT
name	VARCHAR2(100)	NOT NULL
email	VARCHAR2(100)	UNIQUE

phone	VARCHAR2(15)	NOT NULL
address	VARCHAR2(255)	NULLABLE

2. Product Table

Column	Data Type	Constraints
product_id	NUMBER (PK)	PRIMARY KEY
name	VARCHAR2(100)	NOT NULL
category	VARCHAR2(50)	NOT NULL
price	DECIMAL(10,2)	NOT NULL
stock_quantity	INT	NOT NULL

3. Order_Details Table

Column	Data Type	Constraints
order_id	NUMBER (PK)	PRIMARY KEY
customer_id	NUMBER (FK)	FOREIGN KEY REFERENCES Customer(customer_id)
order_date	DATE	NOT NULL
total_amount	DECIMAL(10,2)	NOT NULL

4. Order_Item Table

Column	Data Type	Constraints
order_id	NUMBER (FK)	FOREIGN KEY REFERENCES Order_Details(order_id)

product_id	NUMBER (FK)	FOREIGN KEY REFERENCES Product(product_id)
quantity	INT	NOT NULL
subtotal	DECIMAL(10,2)	NOT NULL

5. Employee Table

Column	Data Type	Constraints
employee_id	NUMBER (PK)	PRIMARY KEY
name	VARCHAR2(100)	NOT NULL
role	VARCHAR2(50)	NOT NULL
salary	DECIMAL(10,2)	NOT NULL
hire_date	DATE	NOT NULL

Examples of Joins

INNER JOIN: Get order details with customer names

```
SELECT o.order_id, c.name, o.order_date, o.total_amount FROM
Order_Details o INNER JOIN Customer c ON o.customer_id =
c.customer_id;
```

```
+----------+------------+------------+--------------+
| order_id | name       | order_date | total_amount |
+----------+------------+------------+--------------+
|        1 | John Doe   | 2024-03-01 |      2000.99 |
|        2 | Jane Smith | 2024-03-02 |       800.50 |
+----------+------------+------------+--------------+
```

INNER JOIN: Retrieve product names and their order quantities

```
SELECT p.name, oi.quantity FROM Order_Item oi INNER JOIN
Product p ON oi.product_id = p.product_id;
```

```
+--------+----------+
| name   | quantity |
+--------+----------+
| Laptop |        1 |
| Phone  |        1 |
+--------+----------+
```

LEFT JOIN: Get all customers and their orders (including those who never ordered)

```
SELECT c.name, o.order_id, o.total_amount FROM Customer c
LEFT JOIN Order_Details o ON c.customer_id = o.customer_id;
```

```
 +------------+----------+--------------+
 | name       | order_id | total_amount |
 +------------+----------+--------------+
 | John Doe   |        1 |      2000.99 |
 | Jane Smith |        2 |       800.50 |
 +------------+----------+--------------+
```

LEFT JOIN: Retrieve all products and their order details (including those not ordered yet)

```
SELECT p.name, oi.quantity FROM Product p
LEFT JOIN Order_Item oi ON p.product_id = oi.product_id;
```

```
+--------+----------+
| name   | quantity |
+--------+----------+
| Laptop |        1 |
| Phone  |        1 |
+--------+----------+
```

RIGHT JOIN: Get all orders with or without employee assigned

```
SELECT o.order_id, e.name AS employee_name FROM
Order_Details o RIGHT JOIN Employee e ON o.customer_id =
e.employee_id;
```

```
+----------+---------------+
| order_id | employee_name |
+----------+---------------+
|        1 | Alice Brown   |
|        2 | Bob White     |
+----------+---------------+
```

RIGHT JOIN: Retrieve employees who processed orders

```
SELECT e.name, o.order_id FROM Employee e RIGHT JOIN
Order_Details o ON e.employee_id = o.customer_id;
```

```
+-------------+----------+
| name        | order_id |
+-------------+----------+
| Alice Brown |        1 |
| Bob White   |        2 |
+-------------+----------+
```

FULL OUTER JOIN: Get all customers and orders (Oracle SQL only)

```
SELECT c.name, o.order_id, o.total_amount FROM Customer c
FULL OUTER JOIN Order_Details o ON c.customer_id =
o.customer_id;
```

CROSS JOIN: Show all possible employee-product assignments

```
SELECT e.name AS employee, p.name AS product FROM Employee e
CROSS JOIN Product p;
```

```
+-------------+---------+
| employee    | product |
+-------------+---------+
| Bob White   | Laptop  |
| Alice Brown | Laptop  |
| Bob White   | Phone   |
| Alice Brown | Phone   |
+-------------+---------+
```

SELF JOIN: Find employees earning more than their colleagues

```
SELECT e1.name AS Employee, e2.name AS Colleague, e1.salary
FROM Employee e1 JOIN Employee e2 ON e1.salary > e2.salary;
```

```
+-------------+-----------+----------+
| Employee    | Colleague | salary   |
+-------------+-----------+----------+
| Alice Brown | Bob White | 50000.00 |
+-------------+-----------+----------+
```

SELF JOIN: Find employees working under the same manager

```
SELECT e1.name AS Employee, e2.name AS Manager FROM Employee
e1 JOIN Employee e2 ON e1.role = 'Cashier' AND e2.role =
'Manager';
```

```
+-----------+-------------+
| Employee  | Manager     |
+-----------+-------------+
| Bob White | Alice Brown |
+-----------+-------------+
```

Joins Tasks

1. Retrieve **customer names** along with their orders.

   SELECT c.name, o.order_id, o.total_amount
   FROM Customer c
   INNER JOIN Order_Details o ON c.customer_id = o.customer_id;

   ```
   +------------+----------+--------------+
   | name       | order_id | total_amount |
   +------------+----------+--------------+
   | John Doe   |        1 |      2000.99 |
   | Jane Smith |        2 |       800.50 |
   +------------+----------+--------------+
   ```

2. Show **product names** and their **order quantities**.

   SELECT p.name, oi.quantity
   FROM Order_Item oi
   INNER JOIN Product p ON oi.product_id = p.product_id;

```
+--------+----------+
| name   | quantity |
+--------+----------+
| Laptop |        1 |
| Phone  |        1 |
+--------+----------+
```

3. List all customers and their orders (**including those who never ordered**).

```
SELECT c.name, o.order_id, o.total_amount
FROM Customer c
LEFT JOIN Order_Details o ON c.customer_id = o.customer_id;
```

```
+------------+----------+--------------+
| name       | order_id | total_amount |
+------------+----------+--------------+
| John Doe   |        1 |      2000.99 |
| Jane Smith |        2 |       800.50 |
+------------+----------+--------------+
```

4. Retrieve **all products and their order details** (**including those not ordered yet**).

```
SELECT p.name, oi.quantity
FROM Product p
LEFT JOIN Order_Item oi ON p.product_id = oi.product_id;
```

```
+--------+----------+
| name   | quantity |
+--------+----------+
| Laptop |        1 |
| Phone  |        1 |
+--------+----------+
```

5. Find employees who have **processed orders**.

```
SELECT e.name, o.order_id
FROM Employee e
RIGHT JOIN Order_Details o ON e.employee_id = o.customer_id;
```

```
+-------------+----------+
| name        | order_id |
+-------------+----------+
| Alice Brown |        1 |
| Bob White   |        2 |
+-------------+----------+
```