Practical 5 Part II

SHIVANSH LOHANI

23070521141

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

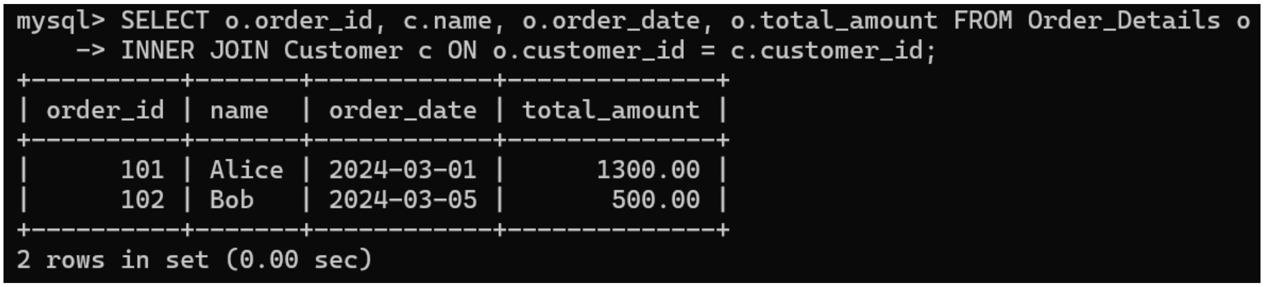
5. Employee Table

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

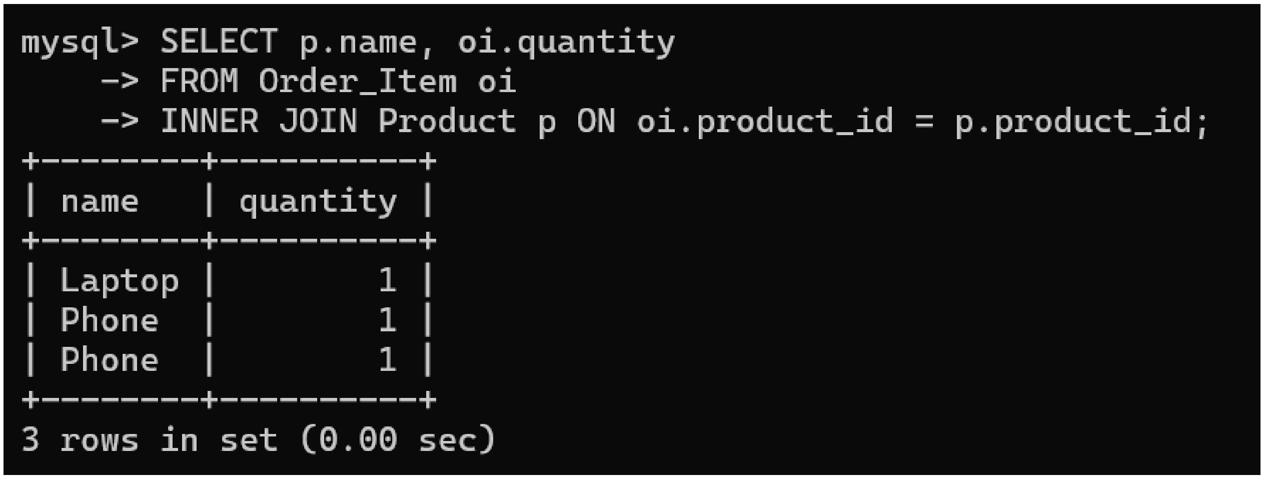
```
SELECT o.order_id, c.name, o.order_date, o.total_amount
FROM Order_Details o
INNER JOIN Customer c ON o.customer_id = c.customer_id;
```



```
mysql> 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 |
+----------+-------+------------+--------------+
|      101 | Alice | 2024-03-01 |      1300.00 |
|      102 | Bob   | 2024-03-05 |       500.00 |
+----------+-------+------------+--------------+
2 rows in set (0.00 sec)
```

INNER JOIN: Retrieve product names and their order quantities

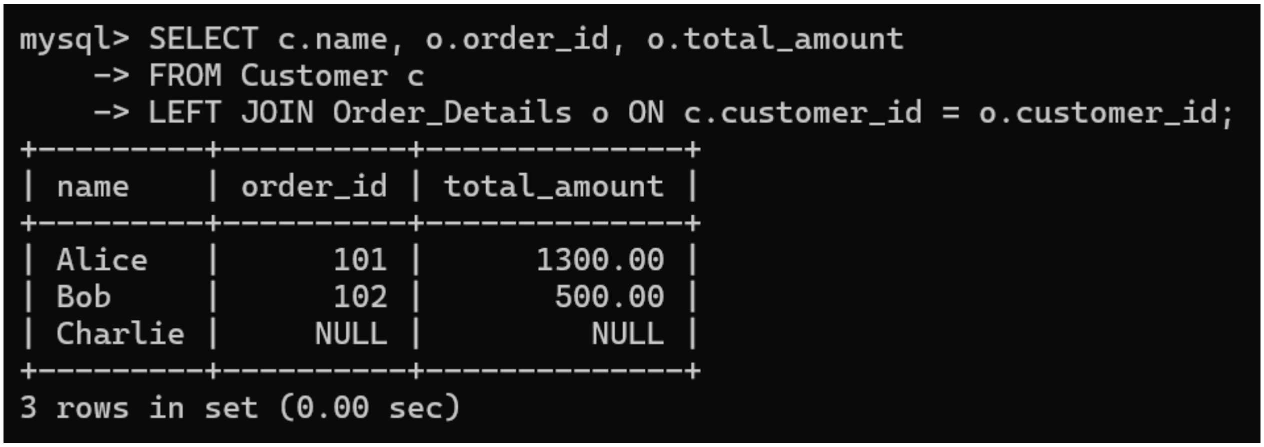
```
SELECT p.name, oi.quantity
FROM Order_Item oi
INNER JOIN Product p ON oi.product_id = p.product_id;
```



```
mysql> 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 |
| Phone  |        1 |
+--------+----------+
3 rows in set (0.00 sec)
```

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

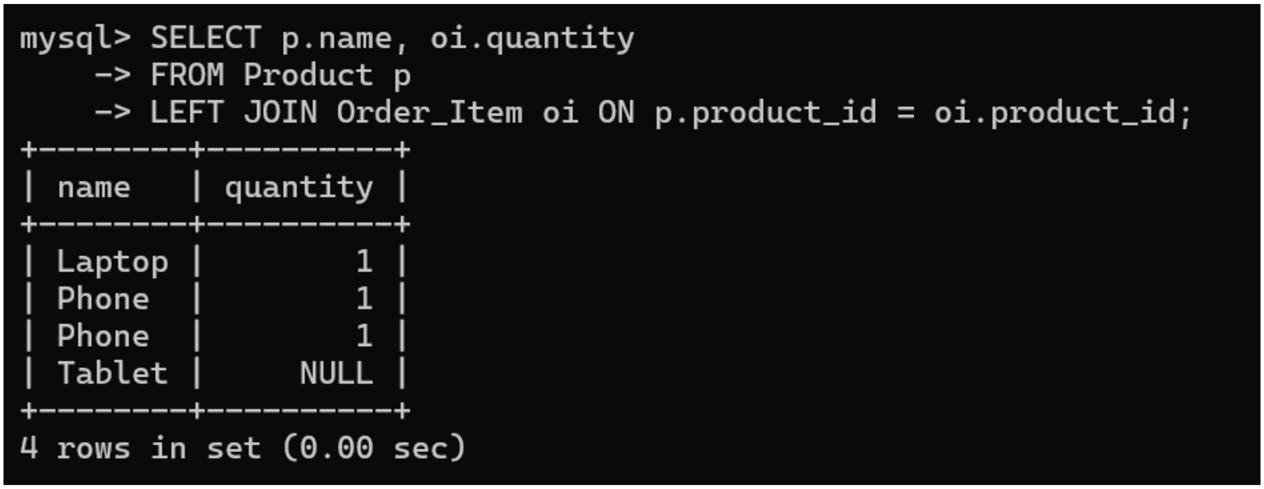
```
SELECT c.name, o.order_id, o.total_amount
FROM Customer c
LEFT JOIN Order_Details o ON c.customer_id = o.customer_id;
```



```
mysql> 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 |
+---------+----------+--------------+
| Alice   |      101 |      1300.00 |
| Bob     |      102 |       500.00 |
| Charlie |     NULL |         NULL |
+---------+----------+--------------+
3 rows in set (0.00 sec)
```

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

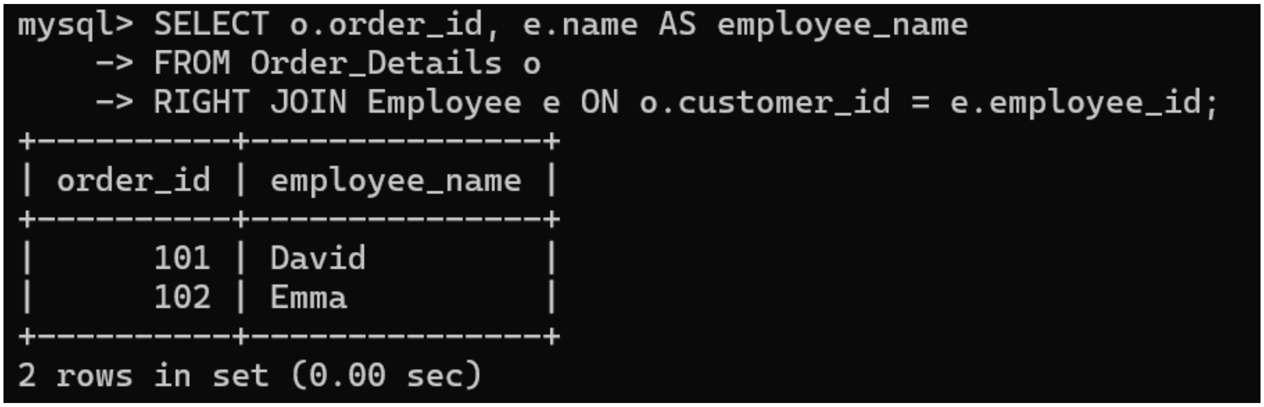
```
SELECT p.name, oi.quantity
FROM Product p
LEFT JOIN Order_Item oi ON p.product_id = oi.product_id;
```



```
mysql> 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 |
| Phone  |        1 |
| Tablet |     NULL |
+--------+----------+
4 rows in set (0.00 sec)
```

RIGHT JOIN: Get all orders with or without employee assigned

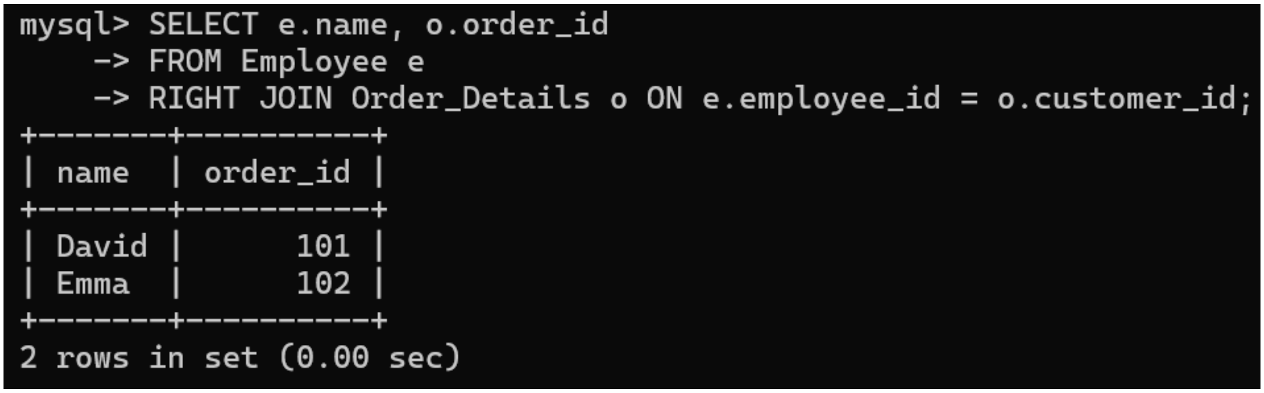
```
SELECT o.order_id, e.name AS employee_name
FROM Order_Details o
RIGHT JOIN Employee e ON o.customer_id = e.employee_id;
```



```
mysql> 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 |
+----------+---------------+
|      101 | David         |
|      102 | Emma          |
+----------+---------------+
2 rows in set (0.00 sec)
```

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;
```



```
mysql> SELECT e.name, o.order_id
    -> FROM Employee e
    -> RIGHT JOIN Order_Details o ON e.employee_id = o.customer_id;
+-------+----------+
| name  | order_id |
+-------+----------+
| David |      101 |
| Emma  |      102 |
+-------+----------+
2 rows in set (0.00 sec)
```

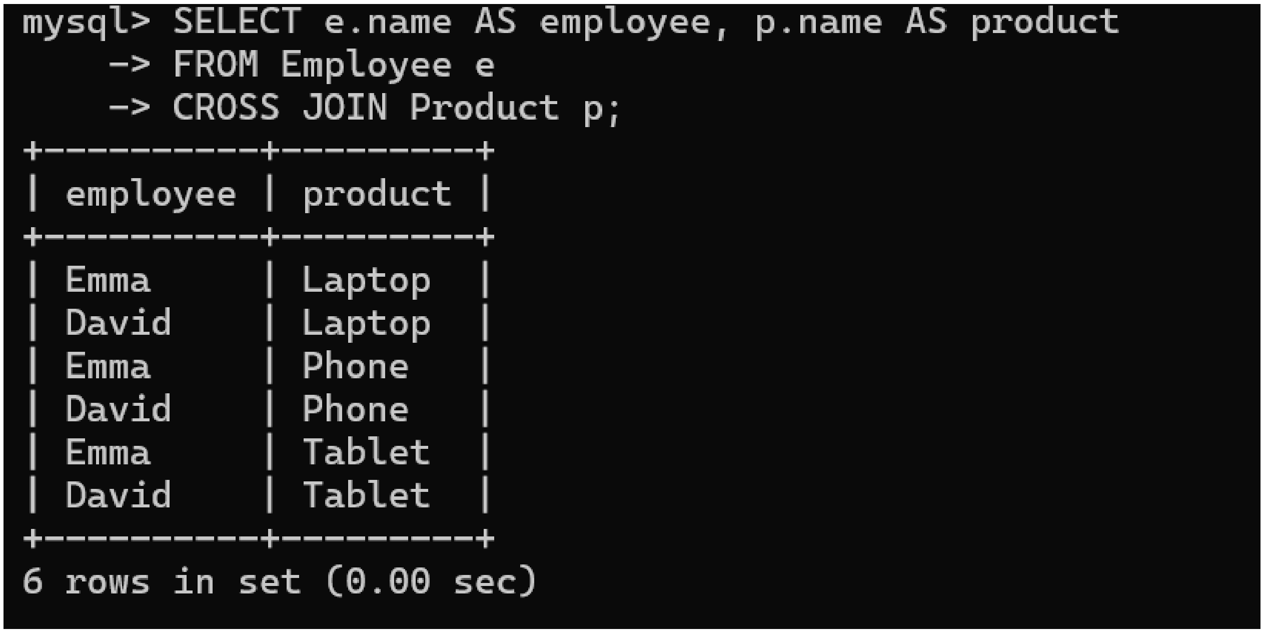
FULL OUTER JOIN: Get all customers and orders (Oracle SQL only)

```
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;
```



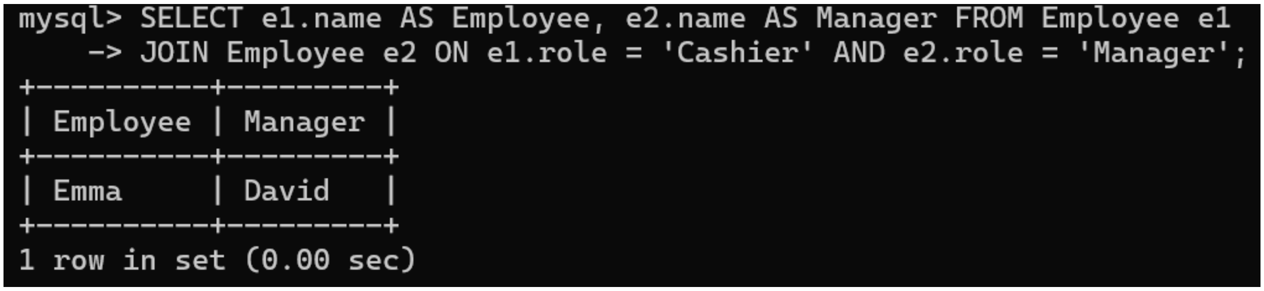
```
mysql> SELECT e.name AS employee, p.name AS product
    -> FROM Employee e
    -> CROSS JOIN Product p;
+----------+---------+
| employee | product |
+----------+---------+
| Emma     | Laptop  |
| David    | Laptop  |
| Emma     | Phone   |
| David    | Phone   |
| Emma     | Tablet  |
| David    | Tablet  |
+----------+---------+
6 rows in set (0.00 sec)
```

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;
```

SELF JOIN: Find employees working under the same manager

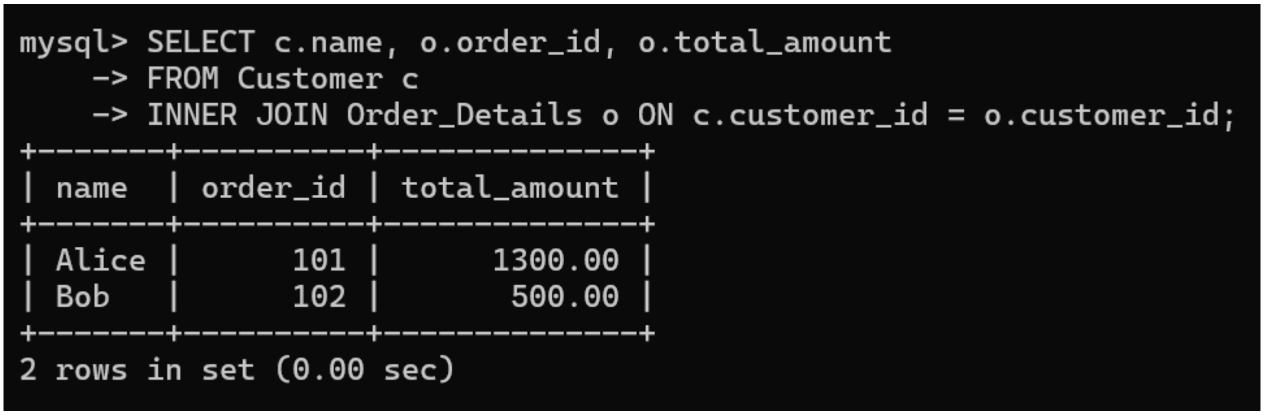
```
SELECT e1.name AS Employee, e2.name AS Manager
FROM Employee e1
JOIN Employee e2 ON e1.role = 'Cashier' AND e2.role =
'Manager';
```



```
mysql> 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 |
+----------+---------+
| Emma     | David   |
+----------+---------+
1 row in set (0.00 sec)
```

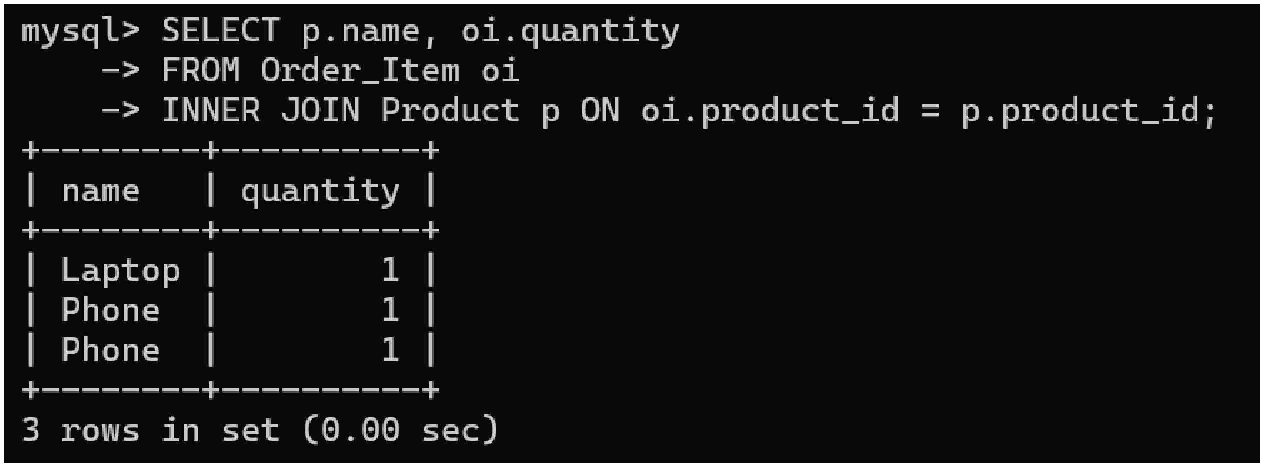
Joins Tasks

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



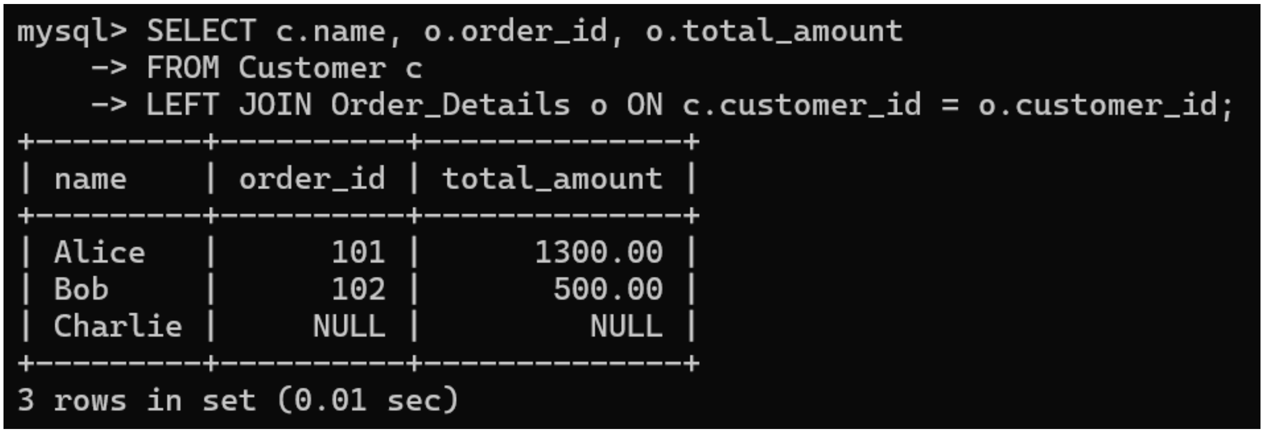
```
mysql> SELECT c.name, o.order_id, o.total_amount
    -> FROM Customer c
    -> INNER JOIN Order_Details o ON c.customer_id = o.customer_id;
+-------+----------+--------------+
| name  | order_id | total_amount |
+-------+----------+--------------+
| Alice |      101 |      1300.00 |
| Bob   |      102 |       500.00 |
+-------+----------+--------------+
2 rows in set (0.00 sec)
```

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



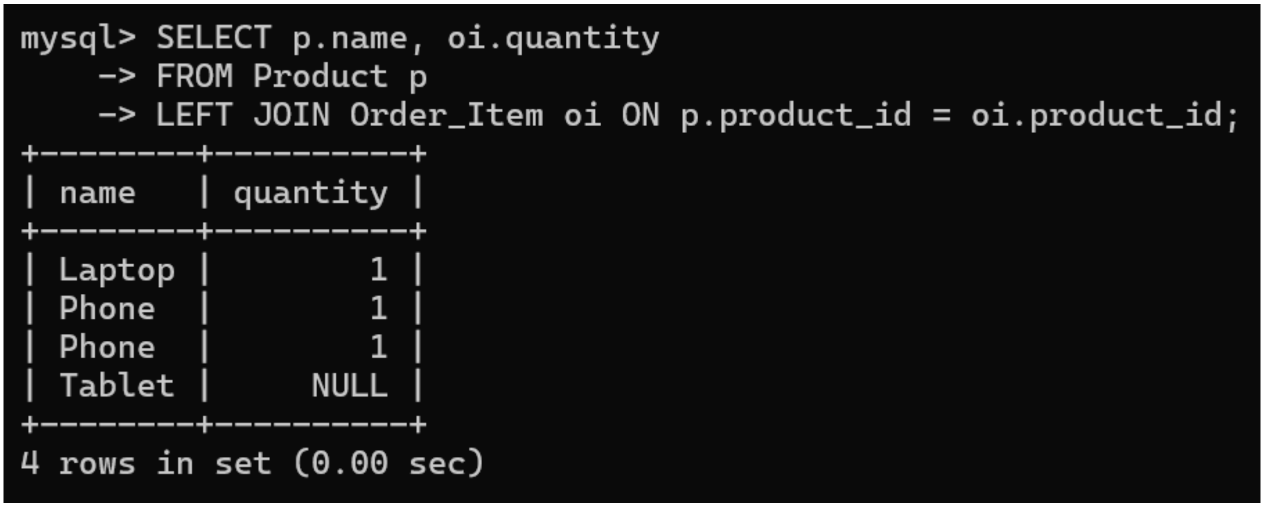
```
mysql> SELECT p.name, oi.quantity
    -> FROM Order_Item oi
    -> INNER JOIN Product p ON oi.product_id = p.product_id;
+--------+----------+
| name   | quantity |
+--------+----------+
| Laptop |        1 |
| Phone  |        1 |
| Phone  |        1 |
+--------+----------+
3 rows in set (0.00 sec)
```

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



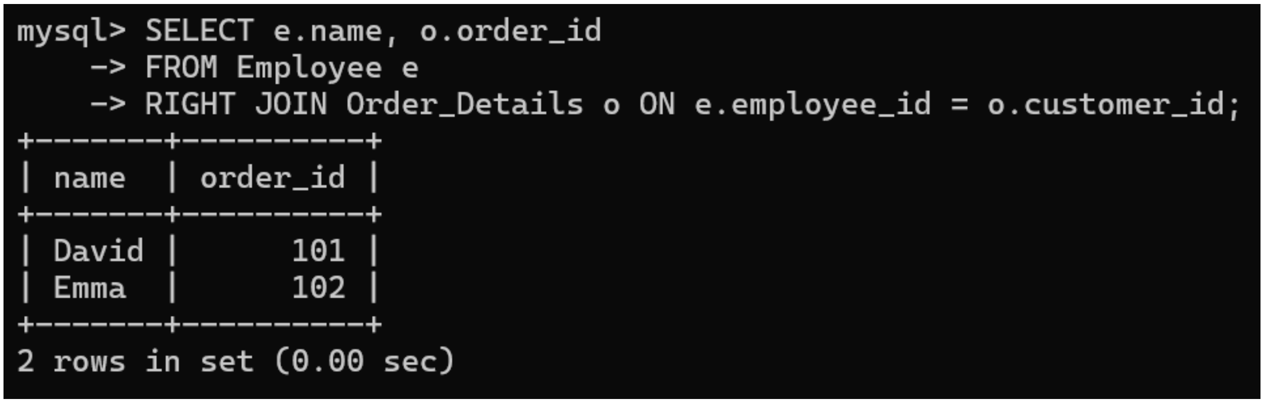
```
mysql> SELECT c.name, o.order_id, o.total_amount
    -> FROM Customer c
    -> LEFT JOIN Order_Details o ON c.customer_id = o.customer_id;
+---------+----------+--------------+
| name    | order_id | total_amount |
+---------+----------+--------------+
| Alice   |      101 |      1300.00 |
| Bob     |      102 |       500.00 |
| Charlie |     NULL |         NULL |
+---------+----------+--------------+
3 rows in set (0.01 sec)
```

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



```
mysql> 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 |
| Phone  |        1 |
| Tablet |     NULL |
+--------+----------+
4 rows in set (0.00 sec)
```

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



```
mysql> SELECT e.name, o.order_id
    -> FROM Employee e
    -> RIGHT JOIN Order_Details o ON e.employee_id = o.customer_id;
+-------+----------+
| name  | order_id |
+-------+----------+
| David |      101 |
| Emma  |      102 |
+-------+----------+
2 rows in set (0.00 sec)
```