Practical S Part II

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

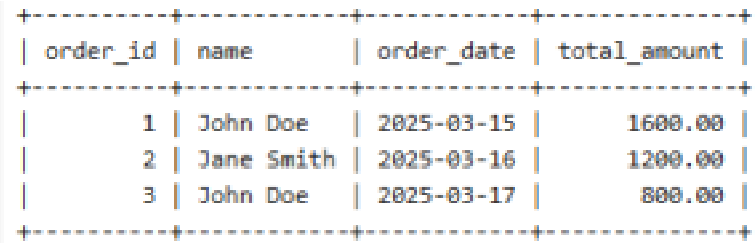
salary DECIMAL(10,2) NOT NULL

hire_date DATE NOT NULL

Examples of Joins

INNER JOIN: Get order details with customer names

```
SELECT o . o rde ra1d , c . name , o . o rde r date , o . t ota 1 amount
FROM 0 rde r Det a11 s o
INNER JOIN Cus tome r c ON o . cust ome ra1d = c . cus tome ra1d ;
```



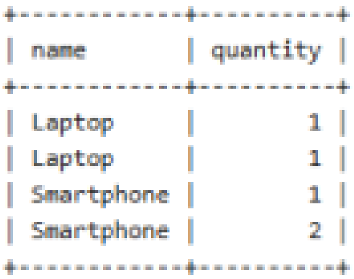
```
+----------+------------+------------+--------------+
| order_id | name       | order_date | total_amount |
+----------+------------+------------+--------------+
|        1 | John Doe   | 2025-03-15 |      1600.00 |
|        2 | Jane Smith | 2025-03-16 |      1200.00 |
|        3 | John Doe   | 2025-03-17 |       800.00 |
+----------+------------+------------+--------------+
```

INNER JOIN: Retrieve product names and their order quantities

```
SELECT p.name, oi.quantity

FROM 0 rde r Item o1

INNER JOIN Product p ON o:l . p roduct :td = p . p roduct :td ;
```



```
+------------+----------+
| name       | quantity |
+------------+----------+
| Laptop     |        1 |
| Laptop     |        1 |
| Smartphone |        1 |
| Smartphone |        2 |
+------------+----------+
```

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

```
SELECT c.name, o.order id, o.total amount
FROM Cusl ome r c
LEFT JOIN 0 rde r Det a:its o ON c . cusl ome re:Id = o . cus tome re:Id ;
```

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;
```

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;
```

RIGHT JOIN: Retrieve employees who processed orders

```
SELECT e . name, o . o rde re:Id
FROM Emp1oyee e
RIGHT JOIN 0 rde r Det a11 s o 0N e . emp1oyee 1d = o . custome ra1d ;
```

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

```
SELECT c . name, o . o rde re:Id ,o . t ot a1 amount
FROM Cus t ome r c
FULL OUTER JOIN 0 rde r Det a11 s o ON c . cus t ome ra1d = o .
cus t ome ra1d ;
```

CROSS JOIN: Show all possible employee-product assignments

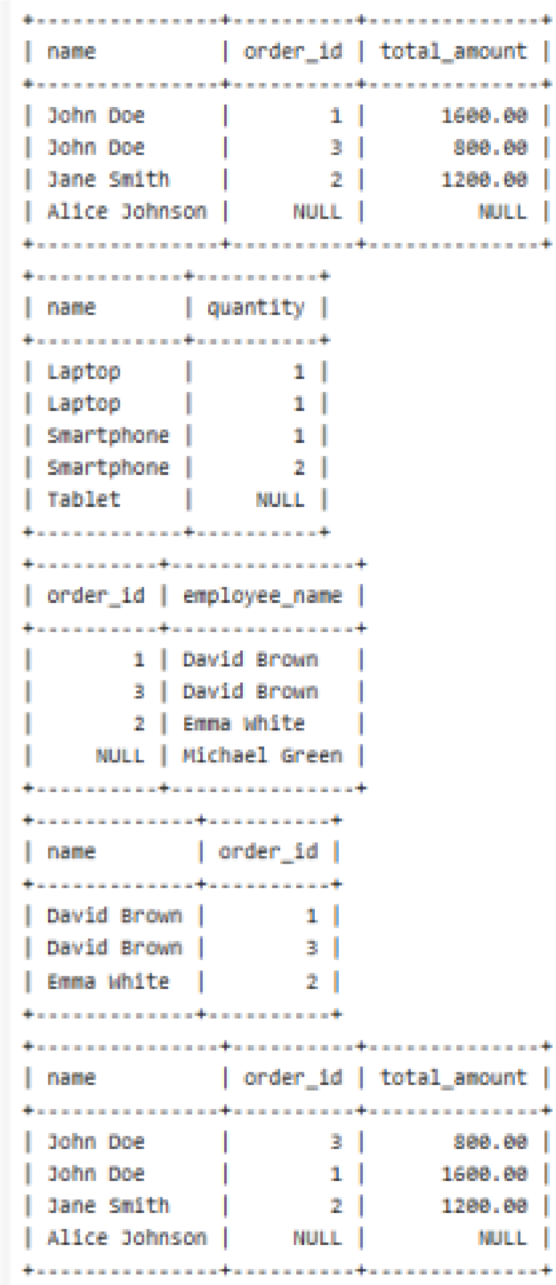
```
SELECT e.name AS employee, p.name AS product
FROM Emp loyee e
CROSS JOIN Product p ;
```

SELF JOIN: Find employees earning more than their colleagues

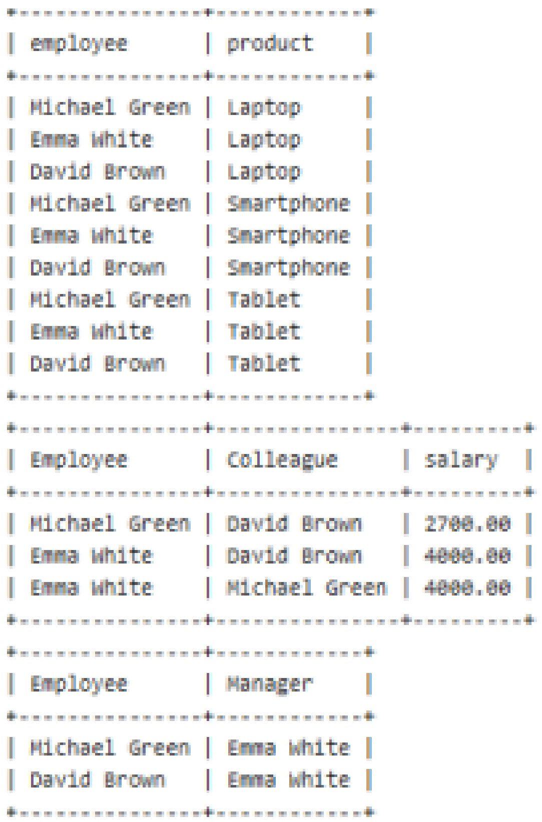
```
SELECT e1.name AS Employee, e2.name AS Colleague,
e1.salary
FROM Emp1oyee e1
JOIN Emp1oyee e2 ON e1 . sa1ary > e2 . sa1ary ;
```

SELF JOIN: Find employees working under the same manager

```
SELECT e1.name AS Employee, e2.name AS Manager
FROM Emp loyee e1
JOIN Emp loyee e2 ON e1 . role = ' Cash1e r ' AND e2. role = '
 Manage r ' ;
```



```
+---------------+----------+--------------+
| name          | order_id | total_amount |
+---------------+----------+--------------+
| John Doe      |        1 |      1600.00 |
| John Doe      |        3 |       800.00 |
| Jane Smith    |        2 |      1200.00 |
| Alice Johnson |     NULL |         NULL |
+---------------+----------+--------------+
+------------+----------+
| name       | quantity |
+------------+----------+
| Laptop     |        1 |
| Laptop     |        1 |
| Smartphone |        1 |
| Smartphone |        2 |
| Tablet     |     NULL |
+------------+----------+
+----------+---------------+
| order_id | employee_name |
+----------+---------------+
|        1 | David Brown   |
|        3 | David Brown   |
|        2 | Emma White    |
|     NULL | Michael Green |
+----------+---------------+
+-------------+----------+
| name        | order_id |
+-------------+----------+
| David Brown |        1 |
| David Brown |        3 |
| Emma White  |        2 |
+-------------+----------+
+---------------+----------+--------------+
| name          | order_id | total_amount |
+---------------+----------+--------------+
| John Doe      |        3 |       800.00 |
| John Doe      |        1 |      1600.00 |
| Jane Smith    |        2 |      1200.00 |
| Alice Johnson |     NULL |         NULL |
+---------------+----------+--------------+
```

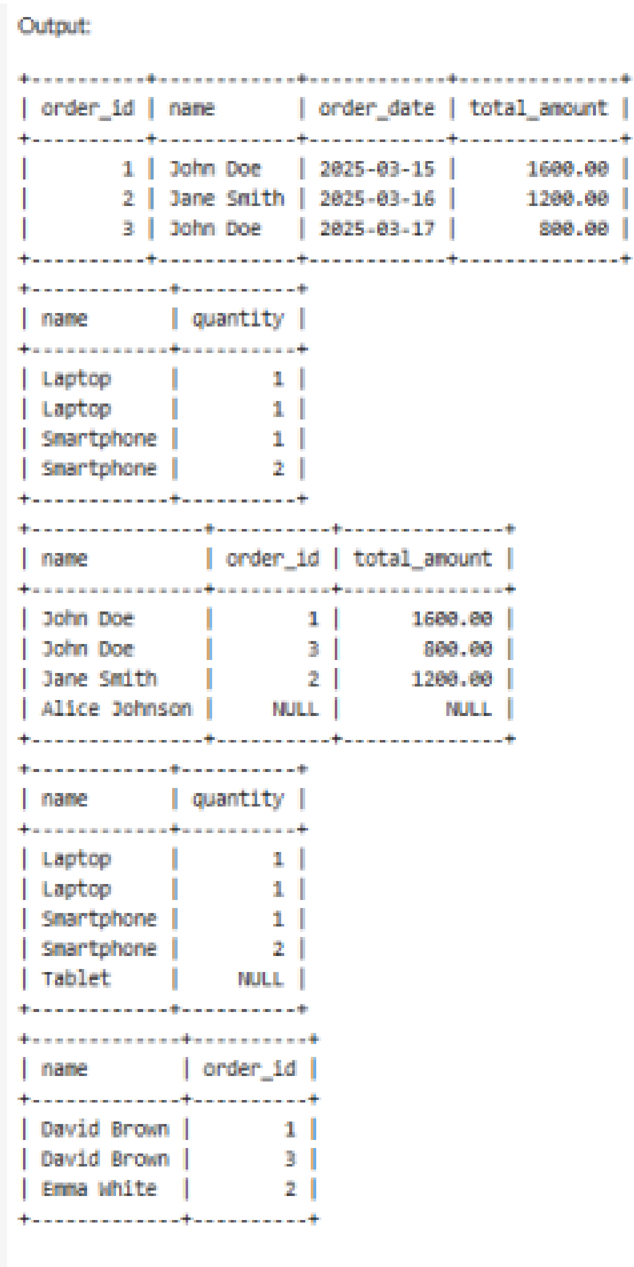


```
+---------------+-----------+
| employee      | product   |
+---------------+-----------+
| Michael Green | Laptop    |
| Emma White    | Laptop    |
| David Brown   | Laptop    |
| Michael Green | Smartphone |
| Emma White    | Smartphone |
| David Brown   | Smartphone |
| Michael Green | Tablet    |
| Emma White    | Tablet    |
| David Brown   | Tablet    |
+---------------+-----------+
+---------------+---------------+---------+
| Employee      | Colleague     | salary  |
+---------------+---------------+---------+
| Michael Green | David Brown   | 2700.00 |
| Emma White    | David Brown   | 4000.00 |
| Emma White    | Michael Green | 4000.00 |
+---------------+---------------+---------+
+---------------+------------+
| Employee      | Manager    |
+---------------+------------+
| Michael Green | Emma White |
| David Brown   | Emma White |
+---------------+------------+
```

Joins Tasks

1. Retrieve customer names along with their orders.
2. Show product names and their order quantities.
3. List all customers and their orders (including those who never ordered).
4. Retrieve all products and their order details (including those not ordered yet).
5. Find employees who have processed orders.

6.



Output:

```
+----------+------------+------------+--------------+
| order_id | name       | order_date | total_amount |
+----------+------------+------------+--------------+
|        1 | John Doe   | 2025-03-15 |      1600.00 |
|        2 | Jane Smith | 2025-03-16 |      1200.00 |
|        3 | John Doe   | 2025-03-17 |       800.00 |
+----------+------------+------------+--------------+
+------------+----------+
| name       | quantity |
+------------+----------+
| Laptop     |        1 |
| Laptop     |        1 |
| Smartphone |        1 |
| Smartphone |        2 |
+------------+----------+
+---------------+----------+--------------+
| name          | order_id | total_amount |
+---------------+----------+--------------+
| John Doe      |        1 |      1600.00 |
| John Doe      |        3 |       800.00 |
| Jane Smith    |        2 |      1200.00 |
| Alice Johnson |     NULL |         NULL |
+---------------+----------+--------------+
+------------+----------+
| name       | quantity |
+------------+----------+
| Laptop     |        1 |
| Laptop     |        1 |
| Smartphone |        1 |
| Smartphone |        2 |
| Tablet     |     NULL |
+------------+----------+
+-------------+----------+
| name        | order_id |
+-------------+----------+
| David Brown |        1 |
| David Brown |        3 |
| Emma White  |        2 |
+-------------+----------+
```