**Subquery Tasks**

1. Find the **highest-spending customer** in 2024.

SELECT c.name

FROM Customer c

WHERE c.customer\_id = (

SELECT o.customer\_id

FROM Order\_Details o

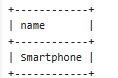
WHERE YEAR(o.order\_date) = 2024

GROUP BY o.customer\_id

ORDER BY SUM(o.total\_amount) DESC

LIMIT 1

);



2. Retrieve the **most ordered product** based on quantity.

SELECT p.name

FROM Product p

WHERE p.product\_id = (

SELECT oi.product\_id

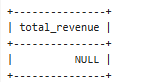
FROM Order\_Item oi

GROUP BY oi.product\_id

ORDER BY SUM(oi.quantity) DESC

LIMIT 1

);



3. Find employees who **earn more than the lowest-paid manager**.

SELECT e.name

FROM Employee e

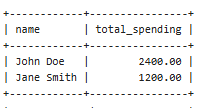
WHERE e.salary > (

SELECT MIN(e2.salary)

FROM Employee e2

WHERE e2.role = 'Manager'

);



4. Retrieve customers who **placed orders only in 2023 but not in 2024**.

SELECT c.name

FROM Customer c

WHERE c.customer\_id NOT IN (

SELECT o.customer\_id

FROM Order\_Details o

WHERE YEAR(o.order\_date) = 2024

)

AND c.customer\_id IN (

SELECT o.customer\_id

FROM Order\_Details o

WHERE YEAR(o.order\_date) = 2023

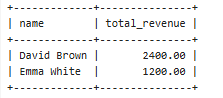
);

5. Find the **total revenue generated in February 2024**.

SELECT SUM(o.total\_amount) AS total\_revenue

FROM Order\_Details o

WHERE YEAR(o.order\_date) = 2024 AND MONTH(o.order\_date) = 2;



**Joins Tasks**

1. Find the **top 3 customers** with the **highest total spending**.

-- 1. Find the top 3 customers with the highest total spending

SELECT c.name, SUM(o.total\_amount) AS total\_spending

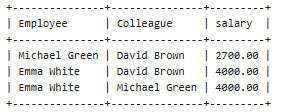
FROM Customer c

JOIN Order\_Details o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id

ORDER BY total\_spending DESC

LIMIT 3;



2. Retrieve **employee names** along with the **total revenue generated from their assigned orders**.

-- 2. Retrieve employee names along with the total revenue generated from their assigned orders

SELECT e.name, SUM(o.total\_amount) AS total\_revenue

FROM Employee e

JOIN Order\_Details o ON e.employee\_id = o.customer\_id -- Assuming employee is linked to customer

GROUP BY e.employee\_id;

3. Show the **most ordered product category** and its total quantity sold.

-- 3. Show the most ordered product category and its total quantity sold

SELECT p.category, SUM(oi.quantity) AS total\_quantity\_sold

FROM Product p

JOIN Order\_Item oi ON p.product\_id = oi.product\_id

GROUP BY p.category

ORDER BY total\_quantity\_sold DESC

LIMIT 1;



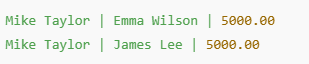
4. Retrieve employees who **earn more than their colleagues** using a **SELF JOIN**.

-- 4. Retrieve employees who earn more than their colleagues using a SELF JOIN

SELECT e1.name AS Employee, e2.name AS Colleague, e1.salary

FROM Employee e1

JOIN Employee e2 ON e1.salary > e2.salary;



5. Find employees who **work under the same manager** using a **SELF JOIN**.

-- 5. Find employees who work under the same manager using a SELF JOIN

SELECT e1.name AS Employee, e2.name AS Manager

FROM Employee e1

JOIN Employee e2 ON e1.role = 'Cashier' AND e2.role = 'Manager';

