TASK QUESTIONS

1. What is a subquery?

A **subquery** is a query written inside another SQL query. It is used to return data that will be used by the main (outer) query.

Example:

SELECT name FROM Employees

WHERE salary > (SELECT AVG(salary) FROM Employees);

---

2. Difference between Subquery and Join

Subquery	Join
A query inside another query	Combines rows from two or more tables
Result of subquery is used by outer query	Returns a combined dataset
Can be slow for large data	Faster because optimized for combining tables
Uses IN, EXISTS, ANY, etc.	Uses JOIN keywords (INNER, LEFT, RIGHT)

---

3. What is a correlated subquery?

A correlated subquery uses a column from the **outer query** inside the **inner query**, meaning it runs once for each row of outer query.

Example:

SELECT name FROM Employees e

WHERE salary > (SELECT AVG(salary) FROM Employees WHERE dept_id = e.dept_id);

---

4. Can subqueries return multiple rows?

Yes, subqueries can return multiple rows when using:

- **IN**

- **ANY**

- **ALL**

- **EXISTS**

Example:

SELECT name FROM Employees

WHERE dept_id IN (SELECT dept_id FROM Departments);

---

5. How does EXISTS work?

EXISTS returns **TRUE** if the subquery returns at least one row. It is used to check for existence.

Example:

SELECT name FROM Employees e

WHERE EXISTS (SELECT * FROM Projects p WHERE p.emp_id = e.emp_id);

---

6. How is performance affected by subqueries?

- Can be **slower** than JOINs, especially correlated subqueries because they execute multiple times.

- Better performance with EXISTS compared to IN for large datasets.

- Optimizers sometimes convert subqueries to joins automatically.

---

7. What is a scalar subquery?

A scalar subquery returns **exactly one value (one row, one column)**.

Example:

SELECT name, salary

FROM Employees

WHERE salary = (SELECT MAX(salary) FROM Employees);

---

8. Where can we use subqueries?

Subqueries can be used in:

- **SELECT**

- **FROM**

- **WHERE**

- **HAVING**

- **WITH**

- **UPDATE/DELETE/INSERT** statements

---

9. Can a subquery be in FROM clause?

Yes. A subquery in the FROM clause is called a **derived table**.

Example:

SELECT dept_id, AVG(salary)

FROM (SELECT * FROM Employees WHERE salary > 50000) AS temp

GROUP BY dept_id;

---

10. What is a derived table?

A derived table is a **subquery used inside the FROM clause** and treated like a temporary table.

Example:

SELECT * FROM (SELECT name, salary FROM Employees) AS empData;