

SUBQUERIES IN SQL

Nested queries in Oracle SQL

Concepts • Syntax • Examples • Practice

Learning Objectives

- Define a subquery and describe why it is useful.
- Write subqueries using comparison operators and IN.
- Recognize single-row, multiple-row, and correlated subqueries.
- Apply subqueries to common employee and department questions.

What Is a Subquery?

A subquery is a SQL query placed inside another SQL statement.

It is also called an inner query or nested query. Its result is used by the outer query to retrieve, update, or delete data.

```
SELECT column_name  
FROM table_name  
WHERE column_name operator (  
    SELECT column_name FROM another_table  
);
```

Typical placement: WHERE, HAVING, FROM, or SELECT (depending on the query).

Example Tables

ORACLE SQL

EMPLOYEES

EMPLOYEE_ID	NAME	SALARY	DEPT_ID
101	Ahmad	30,000	10
102	Bilal	45,000	20
103	Sara	60,000	10
104	Maryam	40,000	30

DEPARTMENTS

DEPT_ID	DEPT_NAME	LOCATION
10	IT	Kabul
20	Finance	Kabul
30	HR	Herat

Use these sample tables in the SQL examples throughout the lesson.

Inner Query and Outer Query

Question: Find employees who earn more than the average salary.

```
SELECT name, salary          -- Outer query
FROM employees
WHERE salary > (
    SELECT AVG(salary)        -- Inner query
    FROM employees
);
```

Inner query result: 43,750

Outer query result: Bilal (45,000) and Sara (60,000)

Single-Row Subquery

A single-row subquery returns one row; it often returns one value.

Example: Find employees earning more than Ahmad.

```
SELECT name, salary
FROM employees
WHERE salary > (
    SELECT salary
    FROM employees
    WHERE name = 'Ahmad'
);
```

The inner query returns 30,000. The outer query returns Bilal, Sara, and Maryam.

Multiple-Row Subquery with IN

Question: Find employees who work in departments located in Kabul.

```
SELECT name, dept_id
FROM employees
WHERE dept_id IN (
    SELECT dept_id
    FROM departments
    WHERE location = 'Kabul'
);
```

Inner query returns department IDs 10 and 20.

Outer query returns Ahmad, Bilal, and Sara.

Using ANY and ALL

ANY and ALL compare a value with the set returned by a subquery.

ANY — at least one comparison is true

```
SELECT name, salary, dept_id
FROM employees
WHERE salary > ANY (
    SELECT salary
    FROM employees
    WHERE dept_id = 10
);
```

For dept 10, salaries are 30,000 and 60,000.

> ANY means greater than at least one value; > ALL means greater than both values.

ALL — every comparison is true

```
SELECT employee_id,
       employee_name,
       department_id,
       salary
FROM employees
WHERE salary > ALL (
    SELECT salary
    FROM employees
    WHERE department_id = 10
);
```

Common Mistakes

- Using = when a subquery can return multiple rows; use IN, ANY, or ALL when appropriate.
- Returning more than one column where a single value is required.
- Forgetting parentheses around the subquery.
- Comparing incompatible data types.
- Using a subquery that returns no rows or unexpected NULL values without considering the result.

Practice Questions

- 1. Write a query to find employees whose salary is below the average salary.
- 2. Find employees who work in the HR department using a subquery.
- 3. Find employees who earn the highest salary.
- 4. Find employees whose salary is higher than the average salary in their own department.
- 5. Identify which questions need a single-row subquery and which need a multiple-row or correlated subquery.

Practice — Sample Answers

ORACLE SQL

1. Below average salary

```
SELECT name, salary
FROM employees
WHERE salary < (SELECT AVG(salary) FROM employees);
```

2. Employees in HR

```
SELECT name
FROM employees
WHERE dept_id = (
    SELECT dept_id FROM departments WHERE dept_name='HR'
);
```

3. Highest-paid employee(s)

```
SELECT name, salary
FROM employees
WHERE salary = (SELECT MAX(salary) FROM employees);
```

4. Above department average

```
SELECT e.name, e.salary
FROM employees e
WHERE e.salary > (
    SELECT AVG(e2.salary) FROM employees e2
    WHERE e2.dept_id = e.dept_id
);
```

Note: These examples use Oracle SQL syntax and the sample tables shown earlier.

Key Takeaways

- A subquery is a query inside another SQL statement.
- The inner query supplies a result to the outer query.
- Use comparison operators for single-row results and IN / ANY / ALL for multiple-row results.
- Correlated subqueries refer to values from the outer query.
- Test queries carefully and consider NULL values.

Questions and discussion