What is a Subquery?

A **subquery** is a **query inside another query**.

Think like this:

- First, **inner query** gives an answer
- Then, **outer query** uses that answer

That's why subquery is also called **nested query**.

Question:

Find students who scored **more than the class average**.

Steps:

1. First, calculate **average marks**
2. Then, compare each student's marks with that average

This **two-step thinking** is exactly what a **subquery** does.

🔹 Types of Subqueries (Overview)

1. **Single-row subquery**
   - Returns **only one value**
2. **Multiple-row subquery**
   - Returns **many values**
3. Multiple-column subquery
4. Scaler Subquery
5. Correlated subquery

1.Single-Row Subquery (One Value)

Meaning

- Subquery returns **only ONE value**
- Used with **WHERE clause**
- Uses **normal comparison operators**

Operators Used

Operator	**Meaning**
=	equal
>	greater than
<	less than
>=	greater than or equal
<=	less than or equal
<>	not equal

Example Table: students

student_id	name	marks
1	Rahul	65
2	Asha	78
3	Neha	90
4	Amit	55

Example 1: Marks greater than average

Step 1: Find average marks

```
SELECT AVG(marks) FROM students;
```

➞ Output: **72**

Step 2: Use this value in main query

```
SELECT *

FROM students

WHERE marks > (SELECT AVG(marks) FROM students);
```

Result:

name	**marks**
Asha	78
Neha	90

Subquery gives **one value (72)** → so **single-row subquery**

Example 2: Student with highest marks

Logic:

- First find **maximum marks**
- Then find student having those marks

```
SELECT *

FROM students

WHERE marks = (SELECT MAX(marks) FROM students);
```

Subquery returns **only one value** → MAX(marks)

Example 3: Salary less than average salary

```
SELECT *

FROM employees

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

Works because subquery gives **one value**

2.Multiple-Row Subquery (Many Values)

Meaning

- Subquery returns **more than one row**
- Cannot use =, >, <
- Uses **special operators**

Operators Used in Multiple-Row Subquery

Operator	**Meaning (Simple)**
IN	matches **any value** in list
NOT IN	does **not** match values
ANY	condition true for **at least one** value
ALL	condition true for **all** values

Example Tables

department

dept_id	**dept_name**	**city**

10	IT	Pune
20	HR	Mumbai
30	Sales	Pune

student

id	**name**	**dept_id**
1	Ravi	10
2	Neha	20
3	Amit	30

Example 1: Students from Pune departments (IN)

Step 1: Find departments in Pune

```
SELECT dept_id

FROM department

WHERE city = 'Pune';
```

➞ Output:

```
10

30
```

Step 2: Main query

```
SELECT *

FROM student

WHERE dept_id IN (

   SELECT dept_id

   FROM department

   WHERE city = 'Pune'

);
```

Subquery returns **many values (10, 30)**
So we use **IN**

Example 2: Students NOT from Pune (NOT IN)

```
SELECT *

FROM student
```

```
WHERE dept_id NOT IN (
   SELECT dept_id
   FROM department
   WHERE city = 'Pune'
);
```

✓ Gives students from **Mumbai**

Example 3: Salary greater than ANY salary of HR dept

Meaning of ANY:

Greater than **at least one** value

```
SELECT *
FROM employees
WHERE salary > ANY (
   SELECT salary
   FROM employees
   WHERE department = 'HR'
);
```

Even if salary is greater than **one HR employee**, it will show

---

Example 4: Salary greater than ALL HR salaries

Meaning of ALL:

Greater than **every value**

```
SELECT *
FROM employees
WHERE salary > ALL (
   SELECT salary
   FROM employees
   WHERE department = 'HR'
);
```

Salary must be **highest of all HR salaries**

Difference Between Subquery and Join (Easy)

Subquery	**Join**
Query inside query	Combine tables
Easy to understand	Slightly complex
Good for beginners	Faster for large data
Step-by-step logic	Direct connection