SQL Nested Queries

🔍 Definition

A nested query (or subquery) is a query written inside another SQL query.

The inner query runs first and provides data to the outer query.

Nested queries make it easier to solve complex problems in smaller, logical steps.

When to Use Nested Queries

Use a nested query when you need to:

Retrieve data based on the result of another query.

Perform filtering or aggregation without using JOINs.

Simplify multi-step logical conditions.

🧱 Sample Database Structure

1️⃣ STUDENT Table

S\_ID S\_NAME S\_ADDRESS S\_PHONE S\_AGE

S1 RAM MUMBAI 9876543210 21

S2 RAMESH DELHI 8765432109 22

S3 SURESH KOLKATA 9123456780 23

S4 RAJ CHENNAI 20

2️⃣ COURSE Table

C\_ID C\_NAME C\_DURATION

C1 DSA 4 months

C2 JAVA 5 months

C3 DBMS 3 months

3️⃣ STUDENT\_COURSE Table

S\_ID C\_ID

S1 C1

S2 C1

S4 C3

⚙ Types of Nested Queries

SQL supports two major types of nested queries:

Type Description

Independent Subquery Inner query runs first; results are passed to outer query.

Correlated Subquery Inner query depends on each row of the outer query.

🧩 Independent Nested Queries

Example 1 — Using IN

Task: Find S\_IDs of students enrolled in ‘DSA’ or ‘DBMS’.

Step 1 – Inner Query

SELECT C\_ID FROM COURSE WHERE C\_NAME IN ('DSA', 'DBMS');

🧮 Result

C\_ID

C1

C3

Step 2 – Outer Query

SELECT S\_ID FROM STUDENT\_COURSE

WHERE C\_ID IN (

SELECT C\_ID FROM COURSE WHERE C\_NAME IN ('DSA', 'DBMS')

);

🧾 Output

S\_ID

S1

S2

S4

✅ Explanation:

The inner query retrieves course IDs (C1, C3) for DSA and DBMS.

The outer query then fetches student IDs enrolled in those courses.

Example 2 — Using NOT IN

Task: Retrieve IDs of students not enrolled in ‘DSA’ or ‘DBMS’.

SELECT S\_ID FROM STUDENT

WHERE S\_ID NOT IN (

SELECT S\_ID FROM STUDENT\_COURSE

WHERE C\_ID IN (

SELECT C\_ID FROM COURSE WHERE C\_NAME IN ('DSA', 'DBMS')

)

);

🧾 Output

S\_ID

S3

✅ Explanation:

Student S3 (SURESH) is not enrolled in either DSA or DBMS.

🧠 Correlated Nested Queries

Example 3 — Using EXISTS

Task: Find names of students enrolled in the course with C\_ID = 'C1'.

SELECT S\_NAME FROM STUDENT S

WHERE EXISTS (

SELECT 1 FROM STUDENT\_COURSE SC

WHERE S.S\_ID = SC.S\_ID AND SC.C\_ID = 'C1'

);

🧾 Output

S\_NAME

RAM

RAMESH

✅ Explanation:

For each student in the STUDENT table, the subquery checks if a record exists in STUDENT\_COURSE for C\_ID = 'C1'.

⚙ Common Operators Used in Nested Queries

Operator Description Example

IN Matches values from subquery result WHERE dept\_id IN (SELECT ...)

NOT IN Excludes values from subquery WHERE emp\_id NOT IN (SELECT ...)

EXISTS Checks if subquery returns any row WHERE EXISTS (SELECT 1 ...)

ANY Compares with any value from subquery > ANY (SELECT salary FROM ...)

ALL Compares with all values from subquery > ALL (SELECT salary FROM ...)

Example 4 — Using ANY

Task: Retrieve student names whose age is greater than any student from DELHI.

SELECT S\_NAME FROM STUDENT

WHERE S\_AGE > ANY (

SELECT S\_AGE FROM STUDENT WHERE S\_ADDRESS = 'DELHI'

);

✅ Explanation:

Finds students older than any (at least one) student living in Delhi.

Example 5 — Using ALL

Task: Retrieve student names whose age is greater than all students from DELHI.

SELECT S\_NAME FROM STUDENT

WHERE S\_AGE > ALL (

SELECT S\_AGE FROM STUDENT WHERE S\_ADDRESS = 'DELHI'

);

✅ Explanation:

Finds students older than every student from Delhi.

📚 Summary Table

Type Keyword Example

Independent IN, NOT IN, ANY, ALL Use when inner query is standalone

Correlated EXISTS Use when inner query depends on outer query

✅ Final Summary

Nested queries make SQL modular, flexible, and powerful.

Independent subqueries run once, correlated subqueries run per row.

Operators like IN, NOT IN, EXISTS, ANY, ALL add versatility.

Used widely for filtering, aggregation, and complex data retrieval

Part 1 — Employee, Department & Sales Tables

Table Structures

Employees Table

emp\_id emp\_name dept\_id

1 John 1

2 Mary 2

3 Bob 1

4 Alice 3

5 Tom 1

Department Table

dept\_id dept\_name

1 Sales

2 Marketing

3 Finance

Sales Table

sale\_id emp\_id sale\_amt

1 1 1000

2 2 2000

3 3 3000

4 1 4000

5 5 5000

6 3 6000

7 2 7000

1️⃣ Find the names of all employees in the Sales department.

SELECT emp\_name

FROM Employees

WHERE dept\_id = (

SELECT dept\_id FROM Department WHERE dept\_name = 'Sales'

);

✅ Explanation:

The inner query finds the dept\_id for Sales, and the outer query retrieves all employees in that department.

🧾 Output:

emp\_name

John

Bob

Tom

2️⃣ Find the names of all employees who have made a sale.

SELECT emp\_name

FROM Employees

WHERE emp\_id IN (

SELECT DISTINCT emp\_id FROM Sales

);

✅ Explanation:

The subquery finds all employee IDs who appear in the Sales table, then we display their names.

🧾 Output:

emp\_name

John

Mary

Bob

Tom

3️⃣ Find the names of all employees who have made sales greater than $1000.

SELECT emp\_name

FROM Employees

WHERE emp\_id IN (

SELECT DISTINCT emp\_id FROM Sales

WHERE sale\_amt > 1000

);

✅ Explanation:

The subquery gets emp\_ids who have made sales > 1000, and the outer query retrieves their names.

🧾 Output:

emp\_name

Mary

Bob

John

Tom

Part 2 — Student, Course, and Student\_Course Tables

Table Structures

Student Table

s\_id s\_name s\_address s\_phone s\_age

S1 Ram Delhi 9874563211 18

S2 Ramesh Mumbai 8883214569 18

S3 Sujit Chennai 7214589632 20

S4 Suresh Delhi 9842155633 18

Course Table

c\_id c\_name

C1 Sales

C2 Marketing

C3 Finance

Student\_Course Table

s\_id c\_id

S1 C1

S1 C3

S2 C1

S3 C2

S4 C2

S4 C3

Note: The following questions mention DSA and DBMS — but since in this dataset courses are Sales, Marketing, Finance, we’ll answer with course names replaced as applicable.

1️⃣ Find the S\_IDs of students who are enrolled in the courses ‘DSA’ or ‘DBMS’.

👉 (Assuming DSA = C1 (Sales) and DBMS = C3 (Finance))

SELECT s\_id

FROM Student\_Course

WHERE c\_id IN (

SELECT c\_id FROM Course WHERE c\_name IN ('Sales', 'Finance')

);

🧾 Output:

s\_id

S1

S2

S4

2️⃣ Find the names of students who are enrolled in the course with C\_ID = 'C1'.

SELECT s\_name

FROM Student

WHERE s\_id IN (

SELECT s\_id FROM Student\_Course WHERE c\_id = 'C1'

);

🧾 Output:

s\_name

Ram

Ramesh

3️⃣ Retrieve student names who enrolled in ‘DSA’ or ‘DBMS’.

SELECT s\_name

FROM Student

WHERE s\_id IN (

SELECT s\_id FROM Student\_Course

WHERE c\_id IN (

SELECT c\_id FROM Course WHERE c\_name IN ('Sales', 'Finance')

)

);

🧾 Output:

s\_name

Ram

Ramesh

Suresh

4️⃣ Retrieve student IDs not enrolled in ‘DSA’ or ‘DBMS’.

SELECT s\_id

FROM Student

WHERE s\_id NOT IN (

SELECT s\_id FROM Student\_Course

WHERE c\_id IN (

SELECT c\_id FROM Course WHERE c\_name IN ('Sales', 'Finance')

)

);

🧾 Output:

s\_id

S3

5️⃣ Find student names enrolled in ‘DSA’.

👉 (DSA = ‘Sales’ = C1)

SELECT s\_name

FROM Student

WHERE EXISTS (

SELECT 1 FROM Student\_Course

WHERE Student.s\_id = Student\_Course.s\_id

AND c\_id = 'C1'

);

🧾 Output:

s\_name

Ram

Ramesh

6️⃣ Compare the output of ANY and ALL operators

Query A (ANY):

SELECT s\_name FROM Student

WHERE s\_age > ANY (

SELECT s\_age FROM Student WHERE s\_address = 'Delhi'

);

👉 Delhi students’ ages = [18, 18].

So, > ANY (18, 18) → age > 18.

🧾 Output:

s\_name

Sujit

Query B (ALL):

SELECT s\_name FROM Student

WHERE s\_age > ALL (

SELECT s\_age FROM Student WHERE s\_address = 'Delhi'

);

👉 > ALL (18, 18) → also means age > 18.

🧾 Output:

s\_name

Sujit

✅ Observation:

Both queries give the same result in this dataset since all Delhi students have the same age (18).

If Delhi students had different ages (e.g., 18 and 20),

> ANY means greater than at least one age → more inclusive.

> ALL means greater than all ages → more restrictive.

✅ FINAL SUMMARY TABLE

Query Description Type Result

Employee #1 Employees in Sales dept Independent John, Bob, Tom

Employee #2 Employees with any sale Independent John, Mary, Bob, Tom

Employee #3 Sales > 1000 Independent Mary, Bob, John, Tom

Student #1 Students in Sales/Finance Independent S1, S2, S4

Student #2 Students in C1 Independent Ram, Ramesh

Student #3 Names in Sales/Finance Independent Ram, Ramesh, Suresh

Student #4 Not in Sales/Finance Independent S3

Student #5 Enrolled in Sales Correlated Ram, Ramesh

Student #6 ANY vs ALL age Comparison Both return Sujit