

Sub Queries

Mirunalini.P

SSNCE

March 25, 2022

Table of Contents

1 Nested Queries

2 References

Session Objective

- To learn about subquery
- To learn about Single row and multirow operators
- To learn about correlated subqueries
- To learn about Set operations

At the end of this session, participants will be able to

- To understand Subquery
- To understand Single row and multirow operators
- To understand Correlated subqueries

Nested Queries

Consider the relations:

Employees (ssn, last_name, first_name, salary, d_id)

Department (dept_id, d_name)

Nested Queries

Which employees have salaries greater than 'john' salary?

- ① find what john earns, and then
- ② find Who earns more than that john's amount.

Problem can be sloved by placing one query inside other query

- The **inner query or the subquery** returns a value that is used by the outer query or the main query.
- A subquery is a **SELECT** statement that is within in a clause of another **SELECT** statement.
- Subquery is very useful when we need to select rows from a table with a condition that depends on the data in the table itself.

```
SELECT select_list  
FROM table  
WHERE expr operator ( SELECT select_list FROM table )
```

- The subquery executes once before the main query.
- The result of the subquery is used by the main query.

The subquery can be in a number of SQL clauses, including:

- **The WHERE clause**
- **The HAVING clause**

Operator includes a comparison condition

- Comparison conditions fall into two classes:
- Single-row operators ($>$, $=$, $>=$, $<$, $<>$, $<=$)
- Multiple-row operators (IN, ANY, ALL).

Nested Query - Rules and Types

- Enlclose suqueries in paranthesis
- Place suqueries on right side of the comparison condition
- Use single-row operators with single-row suqueries
- Multiple-row operators with multiple-row suqueries
- **Single-row subqueries:** Queries that return only one row from the inner SELECT statement
- **Multiple-row subqueries:** Queries that return more than one row from the inner SELECT statement
- **Multiple-column Subqueries:** Queries that return more than one column from the inner SELECT statement.

Nested Query (Single Row queries) - Example -1

- Which employees have salaries greater than 'Pat' salary?

Nested Query (Single Row queries) - Example -1

- Which employees have salaries greater than 'Pat' salary?
- `SELECT last_name, salary FROM employees`
`WHERE salary > (SELECT salary FROM employees`
`WHERE last_name = 'Pat');`

Nested Query (Single Row queries) - Example -1

- Which employees have salaries greater than 'Pat' salary?
- `SELECT last_name, salary FROM employees
WHERE salary > (SELECT salary FROM employees
WHERE last_name = 'Pat');`

First_name	Salary
Jonathon	8600
Kimberely	7000
Michael	13000
Shelley	12000
William	8300

Table 1: Table 1

Nested Query (Single Row queries) - Example -2

- Display the employee name whose dept_id is the same as that of employee 141.

Nested Query (Single Row queries) - Example -2

- Display the employee name whose dept_id is the same as that of employee 141.
- `SELECT last_name, d_id FROM employees WHERE d_id = (SELECT d_id FROM employees where ssn = 141)`

Nested Query (Single Row queries) - Example -2

- Display the employee name whose dept_id is the same as that of employee 141.
- `SELECT last_name, d_id FROM employees WHERE d_id = (SELECT d_id FROM employees where ssn = 141)`

FIRST_NAME	DEPARTMENT_ID
Kevin	50
Trenna	50
Curtis	50
Randall	50
Peter	50

Table 2: Output

Nested Query (Single Row queries) - Example-3

- Displays employees whose d_id is the same as that of employee 141 and whose salary is greater than that of employee 143.

Nested Query (Single Row queries) - Example-3

- Displays employees whose d_id is the same as that of employee 141 and whose salary is greater than that of employee 143.
- `SELECT last_name, d_id, salary FROM employees WHERE d_id = (SELECT d_id FROM employees WHERE ssn = 141) AND salary > (SELECT salary FROM employees WHERE ssn = 143)`

FIRST_NAME	DEPARTMENT_ID
Kevin	50
Trenna	50
Curtis	50

Table 3: Output3

Nested Query (Using Group Functions) - Example - 4

- Display the employee last_name, d_id, and salary of all employees whose salary is a minimum among all employees.

Nested Query (Using Group Functions) - Example - 4

- Display the employee last_name, d_id, and salary of all employees whose salary is a minimum among all employees.
- `SELECT last_name, d_id, salary FROM employees WHERE salary =(SELECT MIN(salary) FROM employees)`

FIRST_NAME	DEPARTMENT_ID	SALARY
Peter	50	2500

Table 4: Output 4

Nested Query (Using Group Functions) - Example-5

- Display all the departments that have a minimum salary greater than department 50.

Nested Query (Using Group Functions) - Example-5

- Display all the departments that have a minimum salary greater than department 50.
- `Select d_id, MIN(salary) FROM employees GROUP BY d_id HAVING MIN(salary) > (SELECT MIN(salary) FROM employees WHERE d_id = 50)`

DEPARTMENT_ID	Min_SALARY
110	8300
90	17000
-	7000
10	4400
20	6000
60	4200
80	8600

Table 5: output 5

Nested Query- Example-5

What is wrong in following query?

- `SELECT ssn, last_name FROM employees WHERE salary=(SELECT MIN(salary) FROM employees GROUP BY d_id)`
- Return more than one row
- Use multiple-row comparison operators
- The multiple-row operator expects one or more values

Multiple - Row Subqueries

Operator	Uses
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the sub

Multiple - Row Subqueries

Employee	e_ssn	last_name	First_name	salaryd_id	
	100	Arun	Kumar	3000	10
	101	Ram	Varma	3400	10
	102	Rajesh	Kumar	4400	20
	103	Anil	Gupta	4300	10
	104	Shyam	Kant	4700	20
	105	Raghu	Raman	3000	20

Department	Dept_id	D_Name
	10	Mkt
	20	Sale

Multiple Row Sub Query (using IN) - Example - 6

- Find the employees who earn the same salary as the minimum salary for each department

Multiple Row Sub Query (using IN) - Example - 6

- Find the employees who earn the same salary as the minimum salary for each department
- (SELECT ssn, last_name, salary, d_id FROM employees WHERE) salary **IN** (SELECT MIN(salary) FROM employees GROUP BY d_id);

ssn	last_name	salary	d_id
100	Arun	3000	10
105	Raghu	3000	20

Nested Query (using ANY) - Example - 7

- Find the employees whose salary is greater than or equal to maximum salary of any of the dept

Nested Query (using ANY) - Example - 7

- Find the employees whose salary is greater than or equal to maximum salary of any of the dept
- (SELECT ssn, last_name, salary, d_id FROM employees WHERE) salary \geq **ANY** (SELECT max(salary) FROM employees GROUP BY d_id); -- $>(4300,4700)$

ssn	last_name	salary	d_id
102	Rajesh	4400	20
103	Anil	4300	10
104	Shyam	4700	20

Nested Query (using ANY) - Example - 8

- Find the employees whose salary is greater than maximum salary any of the depts

Nested Query (using ANY) - Example - 8

- Find the employees whose salary is greater than maximum salary any of the depts
- (SELECT ssn, last_name, salary, d_id FROM employees WHERE) salary > ANY (SELECT max(salary) FROM employees GROUP BY d_id); -- >(4300,4700)

ssn	last_name	salary	d_id
102	Rajesh	4400	20
104	Shyam	4700	20

Nested Query (using ALL) - Example - 9

- Find the employees whose salary is greater than or equal to the maximum salary of all the departments

Nested Query (using ALL) - Example - 9

- Find the employees whose salary is greater than or equal to the maximum salary of all the departments
- (SELECT ssn, last_name, salary FROM employees WHERE) salary \geq ALL (SELECT max(salary) FROM employees GROUP BY d_id); -- \geq (4300,4700)

ssn	last_name	salary
104	Shyam	4700

Nested Query (using ALL) - Example - 10

- Find the employees whose salary is greater than the maximum salary of each dept.

Nested Query (using ALL) - Example - 10

- Find the employees whose salary is greater than the maximum salary of each dept.
- (SELECT ssn, last_name, salary FROM employees WHERE) salary > ALL (SELECT max(salary) FROM employees GROUP BY d_id); -- >(4300,4700)

Nested Query (using ALL) - Example - 10

- Find the employees whose salary is greater than the maximum salary of each dept.
- (SELECT ssn, last_name, salary FROM employees WHERE) salary > ALL (SELECT max(salary) FROM employees GROUP BY d_id); -- >(4300,4700)
- No Rows Selected

Correlated Sub Queries

- Correlated subqueries are used for row-by-row processing.
- Each subquery is executed once for every row of the outer query
- The candidate row is obtained from outer query
- The inner query is executed using candidate row value
- Use values from inner query to qualify or disqualify the candidate row

Nested Sub Queries vs Correlated Subqueries

Nested Sub Queries

- The inner query executes first and finds a value
- The outer query executes once, using the value from the inner query.

Correlated Subquery Execution

- Whenever an inner query is referred by an outer query use correlated subqueries.
- Get a candidate row (fetched by the outer query).
- Execute the inner query using the value of the candidate row.
- Use the values resulting from the inner query to qualify or disqualify the candidate.
- Repeat until no candidate row remains.

Correlated Subquery

```
SELECT column1, column2, ...  
FROM   table1 outer  
WHERE  column1 operator  
        (SELECT column1, column2  
         FROM   table2  
         WHERE  expr1 =  
                outer .expr2);
```

The subquery references a column from a table in the parent query.

Correlated Sub Query- Example - 11

- Find all employees who earn more than the average salary in their department

Correlated Sub Query- Example - 11

- Find all employees who earn more than the average salary in their department
- (SELECT l_name, salary, d_id
FROM employee outer WHERE salary >
(SELECT AVG(salary) FROM employees WHERE
d_id=outer.d_id);

The EXISTS operator.

- This operator is frequently used with correlated subqueries to test whether a value retrieved by the outer query exists in the results set of the values retrieved by the inner query.
 - If the subquery returns at least one row, the operator returns TRUE.
 - If the value does not exist, it returns FALSE.
- NOT EXISTS tests whether a value retrieved by the outer query is not a part of the results set of the values retrieved by the inner query.

Correlated Sub Query (Exists) - Example - 12

- List the department names which have atleast one employee

Correlated Sub Query (Exists) - Example - 12

- List the department names which have atleast one employee
- (SELECT d_id, d_name
FROM dept d WHERE *exists*
(SELECT * FROM employees WHERE e.d_id=d.d_id);

Correlated Sub Query (Not Exists) - Example - 12

- List the department names which do not have any employees

Correlated Sub Query (Not Exists) - Example - 12

- List the department names which do not have any employees
- (SELECT d_id, d_name
FROM dept d WHERE *Not exists*
(SELECT * FROM employees WHERE e.d_id=d.d_id);

Reference



Fundamentals of Database systems 7th Edition by Ramez Elmasri.



Oracle 9i SQL