

#PostgreSQL#



Section 7. Subquery

Subquery – write a query nested inside another query.

Correlated Subquery – show you how to use a correlated subquery to perform a query that depends on the values of the current row being processed.

ANY – retrieve data by comparing a value with a set of values returned by a subquery.

ALL – query data by comparing a value with a list of values returned by a subquery.

EXISTS – check for the existence of rows returned by a subquery.

Can You Teach Me above all the topics with Realtime example (create a simple table and insert some values) and show output in Table Format and Also Explain Each topic why this needed, what happen, like syntax and so on.



To explain these PostgreSQL subquery topics effectively, we'll go through each one with a simple example using a sample table and show the output in a table format. Let's start by defining our sample table and inserting some values:

Table Definition and Insertion

We'll create a `employees` table and insert some data into it.

```
sql

CREATE TABLE employees (
  employee_id SERIAL PRIMARY KEY,
  name VARCHAR(50),
  department VARCHAR(50),
  salary DECIMAL(10, 2)
);

INSERT INTO employees (name, department, salary) VALUES
('Alice', 'HR', 55000),
('Bob', 'Engineering', 75000),
('Charlie', 'Marketing', 50000),
('David', 'Engineering', 80000),
('Eve', 'HR', 60000);
```

1. Subquery

A subquery is a query nested inside another query. It is used to retrieve data that will be used in the outer query. Here's an example:

```
sql

-- Subquery Example
SELECT * FROM employees
```

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

Explanation:

- The inner query `SELECT AVG(salary) FROM employees` calculates the average salary.
- The outer query retrieves all employees whose salary is greater than the calculated average.
- This setup helps us filter employees who earn more than the average salary.

employee_id	name	department	salary
2	Bob	Engineering	75000
4	David	Engineering	80000

2. Correlated Subquery

A correlated subquery is a subquery that depends on the values of the current row being processed by the outer query. Here's an example:

```
sql

-- Correlated Subquery Example
SELECT name, salary, (SELECT MAX(salary) FROM employees WHERE department = e.department) AS
max_salary
FROM employees AS e;
```

Explanation:

- The subquery `SELECT MAX(salary) FROM employees WHERE department = e.department` finds the highest salary in the same department for each employee.
- The outer query fetches the employee's name, salary, and the maximum salary in their department.
- This is useful for comparing an employee's salary with the highest salary in their department.

name	salary	max_salary
Alice	55000	60000
Bob	75000	80000
Charlie	50000	60000
David	80000	80000
Eve	60000	60000

3. ANY

The `ANY` keyword allows you to compare a value to a set of values returned by a subquery. Here's an example:

```
sql

-- ANY Example
SELECT name, salary FROM employees
```

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

Explanation:

- The inner query returns all salaries in the Engineering department.
- The outer query retrieves employees whose salary is greater than any of those in the Engineering department.
- This is useful when you want to filter data based on a specific set of values.

name	salary
Bob	75000
David	80000

4. ALL

The `ALL` keyword allows you to compare a value to a list of values returned by a subquery. Here's an example:

```
sql

-- ALL Example
SELECT name, salary FROM employees
WHERE salary > ALL (SELECT salary FROM employees WHERE department = 'HR');
```

Explanation:

- The inner query returns all salaries in the HR department.
- The outer query filters employees whose salary is greater than all salaries in the HR department.
- This helps in identifying employees whose salaries are higher compared to all other salaries in a specific department.

name	salary
Bob	75000
David	80000

5. EXISTS

The `EXISTS` keyword checks whether a subquery returns any rows. Here's an example:

```
sql

-- EXISTS Example
SELECT name FROM employees
WHERE EXISTS (SELECT * FROM employees WHERE department = 'Marketing');
```

Explanation:

- The subquery checks if there are any employees in the Marketing department.

- The outer query returns the names of employees who belong to departments that exist in the table.
- This is useful for checking the existence of rows in a subquery.

name
Alice
Bob
Charlie
David
Eve

These examples demonstrate how subqueries can be used to perform more complex queries in PostgreSQL. They allow for nested querying and comparison with sets of values, which can be crucial in data analysis and reporting tasks.