**SQL EXISTS and ILIKE Clauses in PostgreSQL**

Let's explore simple examples of the EXISTS and ILIKE clauses using a simple employees and departments table. We'll look at how to create the tables (DDL), insert sample data, and use EXISTS and ILIKE in queries.

**1. Creating Tables (DDL)**

We will use two tables, employees and departments.

sql

Copy code

-- Create departments table

CREATE TABLE departments (

department\_id SERIAL PRIMARY KEY,

department\_name VARCHAR(100)

);

-- Create employees table

CREATE TABLE employees (

employee\_id SERIAL PRIMARY KEY,

name VARCHAR(100),

department\_id INT,

salary NUMERIC(10, 2),

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

**2. Inserting Sample Data**

We will insert some sample data into both tables.

sql

Copy code

-- Insert data into departments

INSERT INTO departments (department\_name)

VALUES

('HR'),

('Engineering'),

('Marketing');

-- Insert data into employees

INSERT INTO employees (name, department\_id, salary)

VALUES

('John Doe', 1, 50000),

('Jane Smith', 2, 70000),

('Alice Johnson', 2, 80000),

('Bob Williams', 3, 45000),

('Sara Lee', 1, 60000);

**3. Example of EXISTS Clause**

The EXISTS clause is used to test for the existence of rows in a subquery. It returns TRUE if the subquery returns any rows, otherwise FALSE.

**Query: Find all departments that have employees**

sql

Copy code

SELECT department\_name

FROM departments d

WHERE EXISTS (

SELECT 1

FROM employees e

WHERE e.department\_id = d.department\_id

);

**Explanation:**

* This query will return departments that have at least one employee.
* The EXISTS clause checks whether the subquery (which checks for matching department\_id in the employees table) returns any rows.

**Output:**

| **department\_name** |
| --- |
| HR |
| Engineering |
| Marketing |

**4. Example of ILIKE Clause**

The ILIKE clause is similar to the LIKE clause but performs a case-insensitive pattern match. It is particularly useful when you want to match strings without worrying about case sensitivity.

**Query: Find all employees whose names start with 'j', ignoring case**

sql

Copy code

SELECT name, salary

FROM employees

WHERE name ILIKE 'j%';

**Explanation:**

* This query will return employees whose names start with the letter 'J' (regardless of case).
* % is the wildcard that matches any sequence of characters.

**Output:**

| **name** | **salary** |
| --- | --- |
| John Doe | 50000.00 |
| Jane Smith | 70000.00 |

**5. Combining EXISTS and ILIKE**

You can also combine both EXISTS and ILIKE in a query. For example, finding all departments that have employees whose names start with 'A' (case-insensitive).

**Query: Find departments with employees whose names start with 'A'**

sql

Copy code

SELECT department\_name

FROM departments d

WHERE EXISTS (

SELECT 1

FROM employees e

WHERE e.department\_id = d.department\_id

AND e.name ILIKE 'A%'

);

**Output:**

| **department\_name** |
| --- |
| Engineering |

**Conclusion**

* **EXISTS** is used to check the existence of rows in a subquery, often to filter results based on whether related data exists.
* **ILIKE** is a case-insensitive variant of LIKE, allowing for flexible string matching.

These clauses make SQL queries more powerful and flexible, especially in scenarios involving complex conditions or case-insensitive searches.

4o