### **Lab 17- Character Functions in PostgreSQL**

Here is a simple lab exercise on character functions in PostgreSQL that you can try out:

**Create a Sample Table**

Let's start by creating a sample table named "Employees" with some data.

CREATE TABLE Employees (

id SERIAL PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100)

);

INSERT INTO Employees (first\_name, last\_name, email) VALUES

('John', 'Doe', 'john.doe@example.com'),

('Jane', 'Smith', 'jane.smith@example.com'),

('Michael', 'Johnson', 'michael.johnson@example.com'),

('Emily', 'Williams', 'emily.williams@example.com');

select \* from employees;

**Lab Exercise Tasks**

Now, let's perform some operations using character functions:

Retrieve the first and last names in uppercase for all employees.

SELECT UPPER(first\_name) AS upper\_first\_name, UPPER(last\_name) AS upper\_last\_name FROM Employees;

Retrieve the first name in lowercase and the last name in uppercase for all employees.

SELECT LOWER(first\_name) AS lower\_first\_name, UPPER(last\_name) AS upper\_last\_name FROM Employees;

Concatenate the first name and last name with a space for all employees.

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM Employees;

Find the length of the email for all employees.

SELECT LENGTH(email) AS email\_length FROM Employees;

Replace the domain of the email addresses with 'company.com' for all employees.

SELECT REPLACE(email, '@example.com', '@company.com') AS updated\_email FROM Employees;

Trim the leading and trailing spaces from the first name for all employees.

SELECT TRIM(first\_name) AS trimmed\_first\_name FROM Employees;

Find the position of the dot ('.') in the email for all employees.

SELECT POSITION('.' IN email) AS dot\_position FROM Employees;

Feel free to execute these SQL commands in your PostgreSQL database to see the results of each operation. This exercise will help you become more familiar with character functions in PostgreSQL.