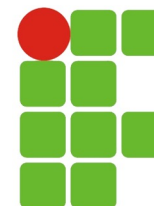


Funções e Stored Procedures

Programação e Administração de Banco de Dados

Prof. Jeferson Queiroga



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Introdução - Função

- Uma função é uma subrotina armazenada no banco de dados que pode realizar operações complexas e retornar um valor.
- Elas são usadas para encapsular a lógica do banco de dados que é repetidamente usada em várias partes de um aplicativo.
- Exemplo de código:

```
CREATE FUNCTION multiply(a integer, b integer)
RETURNS integer AS $$
BEGIN
    RETURN a * b;
END;
$$ LANGUAGE plpgsql;
```



Introdução - Função

- Exemplo de função

```
CREATE OR REPLACE FUNCTION get_employees_by_year(year_in INT)
RETURNS TABLE (
    ID INT,
    Nome TEXT,
    Cargo TEXT,
    Salario NUMERIC,
    Data_Ingresso DATE
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT * FROM funcionarios WHERE EXTRACT(YEAR FROM Data_Ingresso) = year_in;
END; $$

SELECT * FROM get_employees_by_year(2023);
```



Introdução

O que são Stored Procedures?

- Rotinas armazenadas executadas no servidor do banco de dados.
- Escritas em uma linguagem de programação específica (como PL/pgSQL).
- Permite executar um conjunto de instruções SQL de forma sequencial.



Por que usar Stored Procedures?

- **Eficiência:** Stored Procedures permitem que você agrupe várias instruções SQL em uma única chamada de procedimento, reduzindo o tráfego de rede.
- **Segurança:** Ao usar Stored Procedures, você pode restringir o que os usuários podem fazer no banco de dados, garantindo que apenas procedimentos autorizados sejam realizados.
- **Reutilização de código e modularidade:** Uma vez definida, uma Stored Procedure pode ser usada em várias aplicações, proporcionando reutilização de código e modularidade.



Como criar um Procedure

```
CREATE OR REPLACE PROCEDURE update_salary(emp_id int, percent_increase numeric)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE employees SET salary = salary + salary * percent_increase
    WHERE employee_id = emp_id;
END;
$$;
```

Para chamar esta procedure utiliza o comando abaixo:

```
CALL update_salary(100, 0.05);
```



Diferença entre Procedure e Função?

Funções e procedures no PostgreSQL diferem principalmente pelo retorno de valores: funções retornam valores e podem ser usadas em instruções SQL, enquanto procedures **não retornam** valores, são chamadas com **CALL** e são usadas para operações que **modificam os dados**.



Diferença entre Procedure e Função?

Funções e procedures no PostgreSQL diferem principalmente pelo retorno de valores: funções retornam valores e podem ser usadas em instruções SQL, enquanto procedures **não retornam** valores, são chamadas com **CALL** e são usadas para operações que **modificam os dados**.



Exemplo 01

```
CREATE OR REPLACE PROCEDURE update_idade_alunos()
LANGUAGE plpgsql AS $$
DECLARE
    l_aluno RECORD;
    l_data_atual DATE := CURRENT_DATE;
    l_idade INTEGER;
BEGIN
    FOR l_aluno IN SELECT * FROM aluno LOOP
        l_idade := EXTRACT(YEAR FROM l_data_atual) - EXTRACT(YEAR FROM l_aluno.data_nascimento);
        IF EXTRACT(MONTH FROM l_data_atual) < EXTRACT(MONTH FROM l_aluno.data_nascimento) OR
            (EXTRACT(MONTH FROM l_data_atual) = EXTRACT(MONTH FROM l_aluno.data_nascimento) AND
             EXTRACT(DAY FROM l_data_atual) < EXTRACT(DAY FROM l_aluno.data_nascimento)) THEN
            l_idade := l_idade - 1;
        END IF;

        UPDATE aluno SET idade = l_idade WHERE id = l_aluno.id;
    END LOOP;
END;
$$;

CALL update_idade_alunos();
```



Exemplo 02

```
CREATE FUNCTION multiply(a integer, b integer)
RETURNS integer AS $$
BEGIN
RETURN a * b;
END;
$$ LANGUAGE plpgsql;
```



Exemplo 03

```
1  CREATE OR REPLACE PROCEDURE add_aluno (  
2      p_nome VARCHAR(255),  
3      p_data_nascimento DATE,  
4      p_cidade_id INTEGER,  
5      p_universidade_id INTEGER  
6  )  
7  LANGUAGE plpgsql AS $$  
8  BEGIN  
9      INSERT INTO aluno (nome,data_nascimento,cidade_id,universidade_id ) VALUES (  
10         p_nome, p_data_nascimento,p_cidade_id, p_universidade_id );  
11  END;  
12  $$;  
13  
14  CALL add_aluno('Nome do Aluno', '2000-01-01', 1, 1);  
15
```



Exercício de Procedure

1. Criar uma função no seu banco de dados do projeto integrador para retornar o ranking de 10 coisa que mais possui no banco. Exemplo: Os 10 funcionários mais bem pagos.
2. Criar um procedure para alterar alguma tupla no banco de dados.
3. Escreva um procedimento que exclua alguma tupla do seu banco de dados. Você irá passar um id de uma entidade e a procedure ficará responsável de excluir.
4. Escreva um procedimento que mova todos os registros de uma tabela X para uma tabela Y. Exemplo: Transferir da tabela "funcionários" para uma tabela "ex_funcionários". A procedure deve receber como parâmetro o ID do funcionário. Ou seja, irá transferir o funcionário x para a tabela ex_funcionario.

