

Stored procedures, funções e triggers

Vinicius da Silveira Segalin

vinicius.segalin@posgrad.ufsc.br

Sumário

- **Stored procedures**
- Funções
- Triggers
- Exemplos no PostgreSQL
- Exercícios

Stored procedures

- O que é
 - Conjunto de instruções (SQL ou não) executadas no servidor do BD
 - Como se fosse um programa dentro do BD
- Para que serve
 - SQL puro contém limitações
 - Centralizar lógica de negócio (integridade e consistência dos dados)
 - Controle de acesso
 - Reduzir o tráfego de dados na rede
- Como funciona
 - Lógica é colocada dentro da SP
 - SP é executada pelo cliente

Stored procedures

- Algumas vantagens
 - Estruturas de controle
 - Condições (if then else)
 - Laços (while, for)
 - SQL dinâmico
 - Views + parâmetros
 - Podem receber parâmetros de entrada e retornar resultados
- Principal desvantagem
 - Cada SGBD tem sua linguagem

Sumário

- Stored procedures
- **Funções**
- Triggers
- Exemplos no PostgreSQL
- Exercícios

Funções

- O mesmo que Stored Procedures, porém...

Stored Procedures	Funções
Podem retornar nenhum, um ou múltiplos valores	Devem retornar um único valor
Podem ter parâmetros de entrada e saída	Apenas parâmetros de entrada
Podem manipular dados	Não podem manipular dados
Não podem ser usadas em instruções SQL	Podem ser usadas em instruções SQL

Para o PostgreSQL é tudo igual!



Sumário

- Stored procedures
- Funções
- **Triggers**
- Exemplos no PostgreSQL
- Exercícios

Triggers

- O que é
 - Um gatilho que dispara a execução de uma função
- Para que serve
 - Executar automaticamente uma função sempre que uma operação é realizada
- Como funciona
 - Usuário cria um gatilho em uma tabela para determinada operação
 - Quando a operação é realizada, o gatilho dispara
 - Uma função é executada automaticamente

Triggers

- Operação
 - Insert
 - Update
 - Delete
- Momento
 - Antes (before)
 - Depois (after)
- Nível
 - Registro (row-level)
 - Declaração (statement-level)

Triggers

- Retorno
 - Statement-level e Row-level After: sempre NULL
 - Row-level Before: NULL ou NEW
- Quando usar
 - Statement-level: quando não precisa de informação dos registros
 - Before: realizar alguma operação ou notificação
 - After: notificação ou salvar no log
 - Row-level: quando precisa de informação dos registros
 - Before: checar ou modificar o registro antes que persista no banco
 - After: propagar a mudança para outras tabelas

Sumário

- Stored procedures
- Funções
- Triggers
- **Exemplos no PostgreSQL**
- Exercícios

Exemplos no PostgreSQL

```
CREATE [OR REPLACE] FUNCTION nome (parâmetros)
RETURNS tipo_retorno
AS $$
DECLARE variáveis
BEGIN
corpo
END;
$$ LANGUAGE linguagem;
```

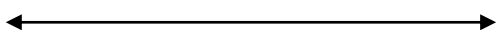
The diagram illustrates the components of a PostgreSQL function definition using arrows:

- nome** (in the original image, this word is red) points to **Nome da função**.
- parâmetros** (in the original image, this word is red) points to **Parâmetros de entrada e saída da função. Ex: IN vl_entrada int, OUT vl_saida int**.
- tipo_retorno** (in the original image, this word is red) points to **Retorno da função. Ex: tipos básicos (integer, boolean...), tabelas**.
- variáveis** (in the original image, this word is red) points to **Variáveis utilizadas na função**.
- corpo** (in the original image, this word is red) points to **Corpo da função, onde fica a lógica**.
- linguagem** (in the original image, this word is red) points to **Linguagem utilizada na função**.

Exemplos no PostgreSQL

- Tipos de variáveis
 - Exemplos:
 - codigo integer
 - quantidade numeric (5)
 - nome varchar (100)
 - pessoa_registro cliente%ROWTYPE
 - pessoa_coluna pessoa.nome%TYPE
- Linguagens
 - SQL
 - PL/pgSQL, PL/Tcl, PL/Perl e PL/Python
 - Permitem utilizar condições e laços

Exemplos no PostgreSQL

```
CREATE FUNCTION add_em (x integer, y integer) RETURNS integer AS $$  
    SELECT x + y;  SELECT $1 + $2;  
$$ LANGUAGE SQL;
```

```
SELECT add_em(1, 2) AS answer;
```

	answer integer
1	3

Exemplos no PostgreSQL

```
CREATE FUNCTION debitar (accountno integer, debit numeric) RETURNS integer AS $$  
  UPDATE bank  
    SET balance = balance - debit  
    WHERE accountno = debitar.accountno  
  RETURNING balance;  
$$ LANGUAGE SQL;
```

	id integer	balance integer
1	1	2000

select debitar (1, 300);

	debitar integer
1	1700

	id integer	balance integer
1	1	1700

Exemplos no PostgreSQL

```
CREATE FUNCTION sum_n_product (x int, y int, OUT sum int, OUT product int)  
AS 'SELECT x + y, x * y'  
LANGUAGE SQL;
```

```
SELECT * FROM sum_n_product(3, 7);
```

	sum integer	product integer
1	10	21

Exemplos no PostgreSQL

```
CREATE FUNCTION getpessoa(id int) RETURNS SETOF pessoa AS $$  
    SELECT * FROM pessoa WHERE id = $1;  
$$ LANGUAGE SQL;
```

```
select * from getpessoa(1);
```

	id integer	nome character varying(100)
1	1	João

Exemplos no PostgreSQL

```
CREATE OR REPLACE FUNCTION maioridade(cod integer)
```

```
RETURNS boolean
```

```
AS $$
```

```
DECLARE maioridade boolean;
```

```
DECLARE idade_retornada integer;
```

```
BEGIN
```

```
SELECT idade into idade_retornada FROM pessoa WHERE id = $1;
```

```
if (idade_retornada >=18) then
```

```
    update pessoa
```

```
    set categoria = 'adulto'
```

```
    where id = $1;
```

```
    maioridade = 1;
```

```
else
```

```
    update pessoa
```

```
    set categoria = 'adolescente'
```

```
    where id = $1;
```

```
    maioridade = 0;
```

```
end if;
```

```
return maioridade;
```

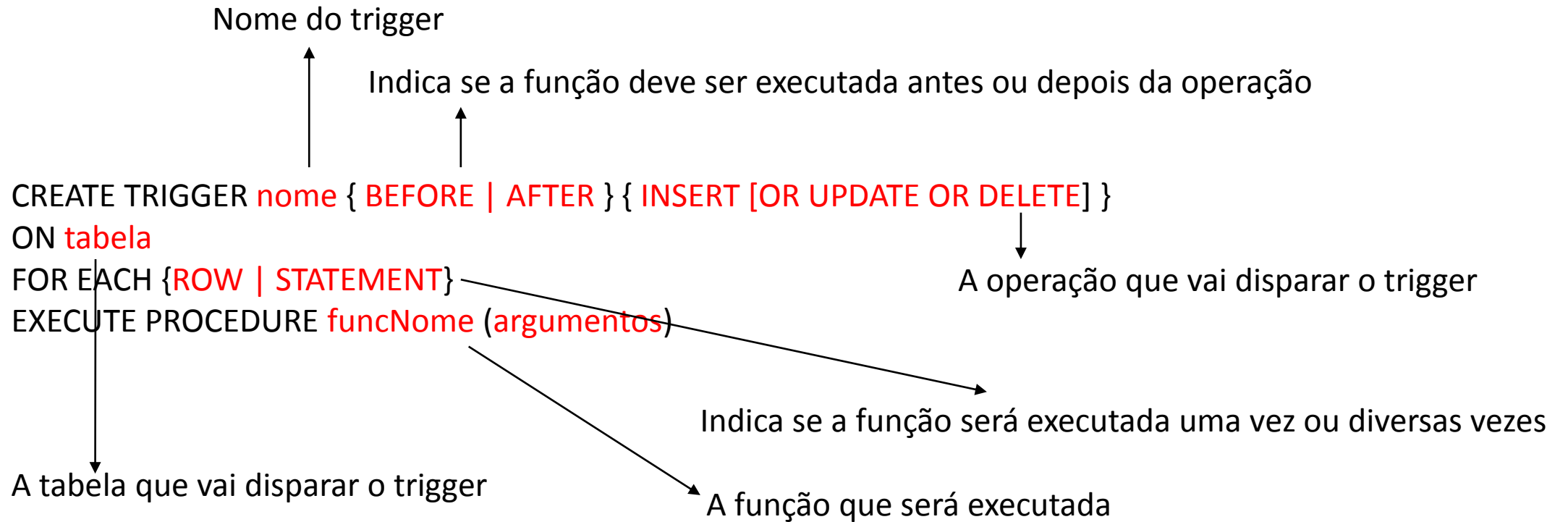
```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
select maioridade (1);
```

	id integer	nome character varying(100)	idade integer	categoria character varying(20)
1	1	João	18	adulto
2	2	Maria	17	
3	3	Carlos	30	

Exemplos em PostgreSQL



Exemplos em PostgreSQL

```
CREATE FUNCTION apenasMaior() RETURNS trigger AS $apenasMaior$  
BEGIN  
  if NEW.idade < 18 then  
    RAISE EXCEPTION 'O usuário % deve ser maior de idade', NEW.nome;  
  end if;  
  return NEW;  
END;  
$apenasMaior$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER apenasMaior BEFORE INSERT ON pessoa  
FOR EACH ROW EXECUTE PROCEDURE apenasMaior();
```

```
insert into pessoa (id, nome, idade) values (4, 'José', 17);
```

```
ERROR: O usuário José deve ser maior de idade  
CONTEXT: PL/pgSQL function apenasmaior() line 4 at RAISE
```

Exemplos em PostgreSQL

```
CREATE FUNCTION maisVelho() RETURNS trigger AS $maisVelho$  
BEGIN  
    if NEW.idade < OLD.idade then  
        RAISE EXCEPTION 'O usuário não pode ficar mais novo';  
    end if;  
    return NEW;  
END;  
$maisVelho$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER maisVelho BEFORE UPDATE ON pessoa  
FOR EACH ROW EXECUTE PROCEDURE maisVelho();
```

```
UPDATE pessoa SET idade = 17 WHERE id = 1;
```

```
ERROR: O usuário não pode ficar mais novo  
CONTEXT: PL/pgSQL function maisvelho() line 4 at RAISE
```

Exemplos em PostgreSQL

```
CREATE OR REPLACE FUNCTION salvar_log() RETURNS trigger AS $salvar_log$  
BEGIN  
    INSERT INTO log VALUES ('O usuário ' || NEW.nome || ' foi salvo');  
    RETURN null;  
END;  
$salvar_log$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER salvar_log AFTER INSERT ON pessoa  
FOR EACH ROW EXECUTE PROCEDURE salvar_log();
```

```
INSERT INTO pessoa (id, nome, idade) VALUES (5, 'Marcos', 20);
```

	comentario character varying(100)
1	O usuário Marcos foi salvo

Sumário

- Stored procedures
- Funções
- Triggers
- Exemplos no PostgreSQL
- **Exercícios**