

PostgreSQL - Funções

Banco de Dados I

Fonte: <https://www.postgresql.org/docs/9.3/static/triggers.html>

<https://www.postgresql.org/docs/9.3/static/sql-createtrigger.html>

PostgreSQL - Trigger

- São implementadas com PL/pgSQL
- Sintaxe do trigger

CREATE TRIGGER nome

tempo (**after,before**) evento (**insert, delete,update**) **ON** tabela tipo (**for each row/statement**)

EXECUTE PROCEDURE nome da funcao que implementa o trigger;

PostgreSQL - Trigger

- Os comandos do trigger são executados por uma função especial que retorna um trigger

Tipo de retorno: trigger

```
create function nome_da_funcao()  
returns trigger () as $$  
--corpo da função
```

Não tem parâmetro

```
$$ language plpgsql
```

Não pode ser implementada com SQL

Trigger

```
create trigger trgalteraqtd after insert on  
funcionario for each row execute procedure  
alteraqtdain();
```

Função executada a partir do Trigger

```
create or replace function alteraqtdain () returns  
trigger as $$  
begin  
    update departamento set qtdef = qtdef+1  
where codigo = new.dept;  
    return null;  
end;  
$$ Language plpgsql;
```

```
funcionário(codigo,nome,dept,salario)  
departamento(codigo,nome,qtdef)  
projeto(codigo,nome)  
alocação(codf,cofp,horas)
```

PostgreSQL - Trigger

- ❑ Crie uma coluna na tabela projeto que deverá armazenar o total de horas de todos os empregados alocados no projeto.
- ❑ Atualize essa coluna de maneira que, para cada projeto, o valor da nova coluna seja igual a soma das horas do projeto correspondente
- ❑ Crie um trigger para atualizar essa coluna quando for inserida uma tupla na tabela alocação

```
funcionário(codigo,nome,dept,salario)  
departamento(codigo,nome,qtdef)  
projeto(codigo,nome)  
alocação(codf,cofp,horas)
```

PostgreSQL - Trigger

```
create or replace function alterahoras() returns trigger as $$  
begin  
    update projeto set totalhoras = totalhoras + new.horas where codigo = new.codp;  
return null;  
end;  
$$ language plpgsql
```

```
create trigger mantemhoras after insert on alocacao  
for each row  
execute procedure alterahoras();
```

PostgreSQL - Trigger

❑ Variáveis especiais

- new: usada em trigger insert e update para referenciar os novos valores de uma coluna na linha modificada
- old: usada no delete e update para referenciar os antigos valores de uma coluna na linha modificada
- TG_WHEN: AFTER ou BEFORE
- TG_LEVEL: ROW ou STATEMENT
- TG_OP: INSERT, DELETE ou UPDATE

create or replace function alteraqtde () returns trigger as \$\$

begin

case-sensitive

if (tg_op = 'INSERT') then

update departamento set qtdef = qtdef+1 where codigo = new.dept;

elseif (tg_op = 'DELETE') then

update departamento set qtdef = qtdef-1 where codigo = old.dept;

elseif ((tg_op = 'UPDATE') and (old.dept != new.dept))then

update departamento set qtdef = qtdef-1 where codigo = old.dept;

update departamento set qtdef = qtdef+1 where codigo = new.dept;

end if;

return null;

end;

\$\$ Language plpgsql;

PostgreSQL - Trigger

```
create trigger trgalteraqtdein  
after insert or delete or  
update on funcionario for  
each row execute procedure  
alteraqtde()
```

PostgreSQL

- ❑ Remover função – trigger
 - DROP TRIGGER nome_trigger ON tabela;
 - ❑ DROP TRIGGER trgalteraqtdein ON funcionario
 - DROP FUNCTION alteraqtde ()
-

PostgreSQL - Trigger

- ❑ Crie um trigger para atualizar a coluna correspondente ao total de horas do projeto sempre que a tabela alocação for atualizada
-

PostgreSQL - Trigger

- ❑ Crie mais duas colunas na tabela projeto: tipo (char(2)) e datatermino (date)
 - ❑ Elabore um trigger que valide a seguinte regra:
 - Caso o tipo do projeto seja 'PD' (prazo determinado), a data de término deve ser obrigatoriamente preenchida. Logo, se um projeto do tipo PD estiver sem data, a inserção ou atualização deve ser interrompida.
-

PostgreSQL - Trigger

- ❑ Qual é o momento desse trigger?
 - ❑ Qual (s) os eventos que podem disparar esse trigger?
 - ❑ Que tipo de trigger é essa?
 - ❑ Caso o tipo seja PD e a data seja nula deve ser gerada uma exceção para que a transação seja interrompida
 - raise exception (encerra transação)
-

PostgreSQL - Trigger

```
create trigger verificatipo before insert or update  
on projeto for each row  
execute procedure validatipo();
```

```
create or replace function validatipo() returns trigger as  
$$  
begin  
if (new.tipo = 'PD' and new.datafim is null) then  
    raise exception 'Projeto PD deve ter data fim';  
end if;  
return new;  
end;  
$$ language plpgsql
```

PostgreSQL

- ❑ Inclua uma coluna na tabela funcionário para armazenar o código do seu gerente
 - ❑ Atualize a tabela funcionário de maneira que todo funcionário fique associado a um gerente
 - ❑ Caso não exista o atributo salário na tabela funcionário, crie o atributo e preencha com um valor qualquer.
 - ❑ Retorne do banco o nome e o salário do funcionário seguido pelo nome e o salário do seu gerente
 - ❑ Caso seja necessário, altere o valor do salário de maneira que o salário de um gerente seja sempre maior que o salário dos seus funcionários
-

PostgreSQL

- ❑ Elabore um trigger para validar a entrada de dados na tabela funcionário de maneira que o valor do salário de um funcionario não seja maior que o valor do seu gerente.
-

PostgreSQL

- ❑ Qual é o momento desse trigger?
 - ❑ Qual (s) os eventos que podem disparar esse trigger?
 - ❑ Que tipo de trigger é essa?
 - ❑ Caso o valor do salário seja maior que o valor do gerente, deve ser gerada uma exceção
 - raise exception (encerra transação)
-

PostgreSQL

```
create trigger salariofuncionario before insert or  
update of salario on funcionario for each row  
execute procedure verificasalario();
```

```
create or replace function verificasalario () returns trigger as $$  
declare salg funcionario.salario%TYPE;  
begin  
    select salario into salg from funcionario where codigo = NEW.codg;  
    if (NEW.salario > salg) then  
        raise exception 'salario do funcionario % e maior que  
gerente',NEW.codigo;  
    end if;  
    return NEW; end;  
$$ Language plpgsql;
```