



TRIGGER

Uma função de gatilho, trigger, é uma ação que o SGBD deve desencadear antes ou depois de um dos seguintes comandos: insert, update ou delete.

Sintaxe:

Create trigger nome_gatilho {before ou after} {evento[or evento]} on table for each {row ou STATEMENT} execute procedure nome_função (v1, v2, ..., vn)

Em gatilho onde tem-se FOR EACH ROW o procedimento em execute procedure é chamado uma vez para cada linha que a operação modifica. Já se tivermos FOR EACH STATEMENT ocorrerá somente uma chamada ao procedimento descrito em execute procedure, não importando quantas linhas sejam modificadas.

VARIÁVEIS DISPONÍVEIS NO AMBIENTE DE PROGRAMAÇÃO – PL

NEW: tipo de dado RECORD; variável contendo a nova linha do banco de dados, para as operações de INSERT/UPDATE nos gatilhos no nível de linha.

OLD: tipo de dado RECORD; variável contendo a antiga linha do banco de dados, para as operações de UPDATE/DELETE nos gatilhos no nível de linha.

TG_NAME: tipo de dado name; variável contendo o nome do gatilho disparado.

TG_WHEN: tipo de dado text; uma cadeia de caracteres contendo BEFORE ou AFTER, dependendo da definição do gatilho.

TG_LEVEL: tipo de dado text; uma cadeia de caracteres contendo ROW ou STATEMENT, dependendo da definição do gatilho.

TG_OP: tipo de dado text; uma cadeia de caracteres contendo INSERT, UPDATE, ou DELETE, informando para qual operação o gatilho foi disparado.

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO POLITÉCNICO
TECNOLOGIA EM GEOPROCESSAMENTO
BANCO DE DADOS ESPACIAL II
PROFª MARINÊZ DA SILVA

TG_RELID: tipo de dado oid; o ID de objeto da tabela que causou o disparo do gatilho.

TG_RELNAME: tipo de dado name; o nome da tabela que causou o disparo do gatilho.

TG_NARGS: tipo de dado integer; o número de argumentos fornecidos ao procedimento de gatilho na instrução CREATE TRIGGER.

TG_ARGV[]: tipo de dado matriz de text; os argumentos da instrução CREATE TRIGGER. O contador do índice começa por 0. Índices inválidos (menor que 0 ou maior ou igual a tg_nargs) resultam em um valor nulo.

EXEMPLO

Criar a tabela func:

```
Create table func(cod integer primary key, nome varchar(30), salario decimal(10,2),  
comissao decimal(10,2));
```

Gatilho que após efetuar uma inclusão, alteração ou exclusão na tabela func armazena dados para auditoria na tabela func_auditoria.

Primeiro – criar a tabela func_auditoria:

```
Create table func_auditoria(operação varchar(1), usuário varchar(30), data timestamp,  
nome_func varchar(40), salario decimal(10,2));
```

Segundo – criar a função para o gatilho

```
Create function processa_audit_func() RETURNS TRIGGER AS $$  
BEGIN  
--Cria uma linha na tabela func_auditoria para refletir a operação realizada na tabela func.  
--Utiliza a variável especial TG_OP para descobrir a operação que está sendo realizada.  
IF (TG_OP = 'DELETE') THEN  
INSERT INTO func_auditoria values('E',user, now(), OLD.nome, OLD.salario);  
RETURN OLD;
```

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO POLITÉCNICO
TECNOLOGIA EM GEOPROCESSAMENTO
BANCO DE DADOS ESPACIAL II
PROFª MARINÊZ DA SILVA

```
ELSEIF (TG_OP='UPDATE') THEN
INSERT INTO func_auditoria values('A', user, now(), OLD.nome, OLD.salario);
RETURN NEW;

ELSEIF (TG_OP='INSERT') THEN
INSERT INTO func_auditoria values('I', user, now(), NEW.nome, NEW.salario);
RETURN NEW;
END IF;
RETURN NULL; -- o resultado é ignorado uma vez que este é um gatilho AFTER
END;
$$ language plpgsql;
```

Terceiro – criar o gatilho

```
CREATE TRIGGER funcionario_audit
AFTER INSERT OR UPDATE OR DELETE ON func
FOR EACH ROW EXECUTE PROCEDURE processa_audit_func();
```

Quando algum usuário fizer um insert, update ou delete na tabela func o SGBD registrará os dados na tabela func_auditoria.

Exemplo:

Inserir um registro na tabela criada

```
Insert into func values(1,'João',1800,500);
```

Se fizermos uma conexão com o SGBD com o usuário postgres e atualizar a tabela func com o comando:

```
Update func set salário = 2000.00 where cod=1;
```

Podemos consultar a tabela func_auditoria para obter como resultado o registro das operações realizadas na tabela func:

```
SELECT * FROM func_auditoria;
```

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO POLITÉCNICO
TECNOLOGIA EM GEOPROCESSAMENTO
BANCO DE DADOS ESPACIAL II
PROFª MARINÊZ DA SILVA

Gatilho para verificar se o nome do funcionário foi informado antes de armazenar na tabela func:

Primeiro – Definição da função do gatilho

```
create function verifica() returns trigger as $$  
begin  
  if new.nome is null then  
    raise exception 'O nome do funcionário não pode ser nulo';  
  end if;  
  return new;  
end;  
$$ language plpgsql;
```

Segundo – Criação do gatilho

```
create trigger novo_func before insert or update on func  
for each row execute procedure verifica();
```

Terceiro – uso do gatilho

```
insert into func(cod,salario,comissao) values(2,4000,700)
```