FATEC

Desenvolvimento de Software Multiplataforma

1º SEMESTRE 2024

BDR - Banco de Dados Relacional

Prof. Me. Eng. Santana

Gatilhos / Triggers

Fatec
Faculdade de Tecnologia

desenvolvimento
de software multiplataforma

Triggers

- Gatilhos/Triggers são procedimentos executados de maneira automática mediante a alguma alteração no banco de dados.
- Em geral são implementados para garantir certas regras de negócio, alerta de alterações ou log de alterações
- Podem ser implementadas para Tabelas ou Views
- Possuem 3 componentes básicos:
  - **Evento**: Alteração que ativa o procedimento; Pode ser antes (BEFORE) ou depois (AFTER) do evento
    - INSERT, UPDATE, DELETE
  - **Condição**: (Opcional) Teste/Verificação executada antes da ação
  - **Ação**: Procedimento/código/SQLs a ser executado

Trigger

CREATE TRIGGER **<nome_trigger>**
<WHEN> <DML*> ON **<nome_tabela>**
FOR EACH ROW
EXECUTE FUNCTION **<nome_funcao> ;**

<WHEN> : BEFORE, AFTER
<DML*> : INSERT, UPDATE, DELETE, TRUNCATE

psql: **\d** <tabela>

Store Procedures/Functions

```
CREATE FUNCTION <nome_function> (<Parametros> <tipo_parametros>)
RETURNS TRIGGER
language plpgsql
AS $$
DECLARE
        variáveis...
BEGIN
        codigos...
        RETURN OLD/NEW;
END $$;
```

Store Procedures/Functions

"Tipo" de variáveis adicionais:

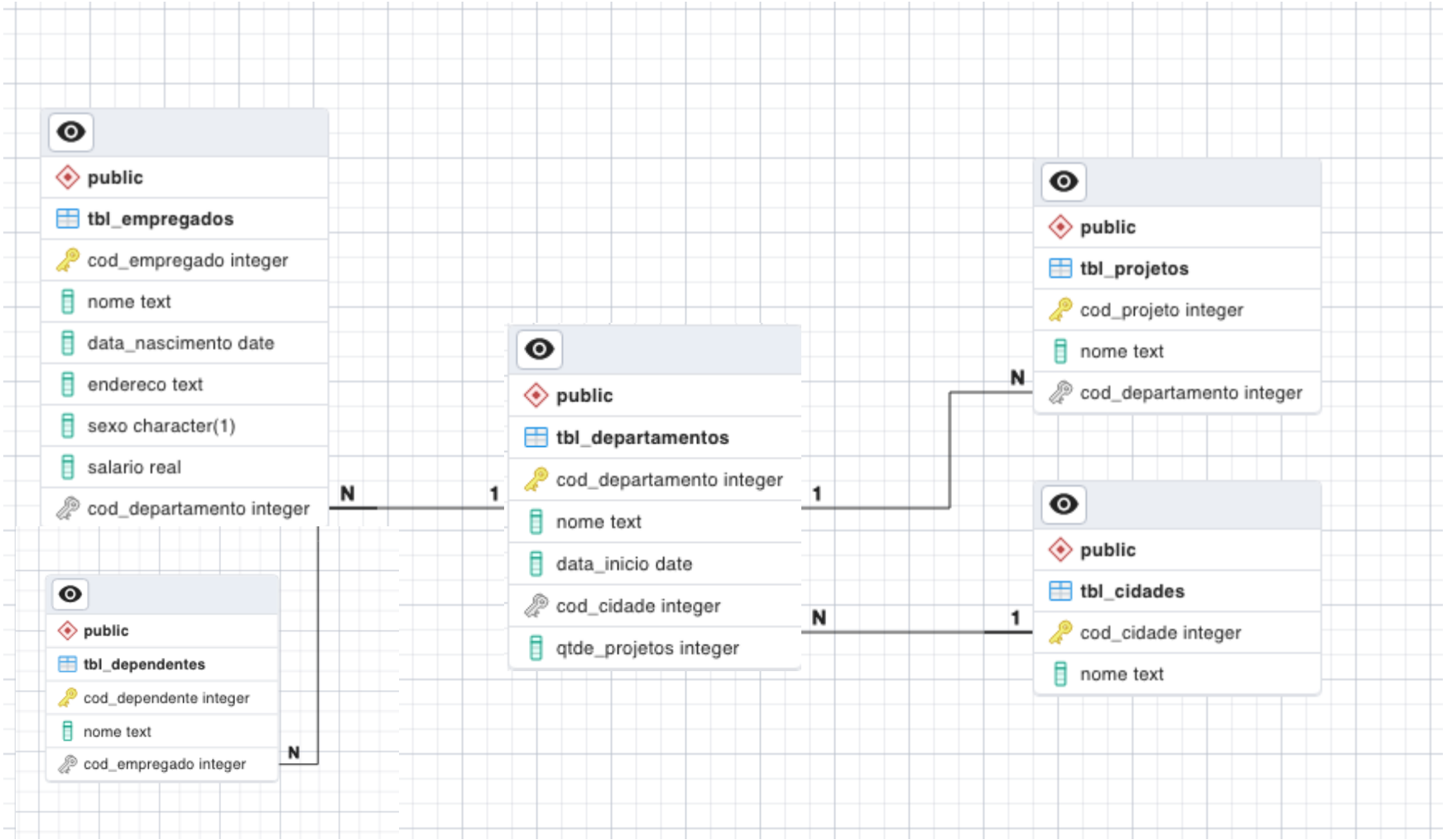
OLD: referencia valor antigo no caso de Update ou Delete

OLD.<coluna>

NEW: referencia valor novo no caso de Insert ou Update.

NEW.<coluna>

LAB - MER


public
tbl_empregados
cod_empregado integer
nome text
data_nascimento date
endereco text
sexo character(1)
salario real
cod_departamento integer
public
tbl_dependentes
cod_dependente integer
nome text
cod_empregado integer
public
tbl_departamentos
cod_departamento integer
nome text
data_inicio date
cod_cidade integer
qtde_projetos integer
public
tbl_projetos
cod_projeto integer
nome text
cod_departamento integer
public
tbl_cidades
cod_cidade integer
nome text
N
1
N
1
N
1
N

Criar TRIGGER

- --crie uma trigger que devera criar uma tabela chamada tbl_projetos_arquivados, caso ela nao exista. tabela deve ter 2 colunas: codigo_projeto e nome.
- --a trigger deve salvar os projeto que forem deletados.

```
CREATE OR REPLACE TRIGGER tr_arquiva_projeto
BEFORE DELETE ON tbl_projetos
FOR EACH ROW
EXECUTE FUNCTION fc_projeto_arquivado();
```

Criar TRIGGER

```
CREATE OR REPLACE FUNCTION fc_projeto_arquivado()
RETURNS TRIGGER
language plpgsql AS
$$
BEGIN
        CREATE TABLE IF NOT EXISTS tbl_projetos_arquivados (cod_projeto int, nome text);
        INSERT INTO tbl_projetos_arquivados VALUES(OLD.cod_projeto, OLD.nome);
        RETURN OLD;
END
$$ ;
```

Criar TRIGGER

- --crie uma trigger que deleta todos os dependentes de um empregado caso ele seja deletado.

```
CREATE OR REPLACE TRIGGER tr_del_dependentes
BEFORE DELETE ON tbl_empregados
FOR EACH ROW
EXECUTE FUNCTION fc_del_dependentes();
```

Criar TRIGGER

```
CREATE OR REPLACE FUNCTION fc_del_dependentes()
RETURNS TRIGGER
language plpgsql AS
$$
BEGIN
        DELETE FROM tbl_dependentes WHERE cod_empregado =
OLD.cod_empregado;
        RETURN OLD;
END
$$ ;
```

Alterações na TRIGGER

É possivel usar ALTER TRIGGER para certas atividades como renomear Trigger porem para alterar a SQL geradora é necessario criar a Trigger novamente usar o “OR REPLACE”

ALTER TRIGGER <nome_trigger> RENAME TO <nome_trigger> ;

ALTER TABLE <nome_tabela> DISABLE TRIGGER <nome_trigger> ;

CREATE OR REPLACE TRIGGER <nome_trigger>

...

DROP Trigger

- **DROP TRIGGER <nome trigger> ON <nome_tabela> ;**

Lab

- Criar banco de dados: bd_aula14
- Executar aula14.sql
- Salvar os SQLs dentro desse arquivo e ao final da aula fazer upload pro github