AULA 16

TRIGGER:
- RETORNO DE FUNÇÕES
- IGNORANDO SENTENÇA DE DISPARO

Professora Luciane Agnoletti

PL/pgSQL E TRIGGERS

- Funções de trigger tem acesso a variáveis especiais que fornecem informações sobre os triggers e permitem que a função manipule dados de tabela.

PL/pgSQL E TRIGGERS

- Variáveis de funções de trigger:

NEW	**RECORD**	Contém a nova linha criada por um INSERT ou UPDATE num trigger row-level. Esta variável é utilizada para fazer modificações na nova linha.
OLD	**RECORD**	Contém a antiga linha processada por um DELETE ou UPDATE num trigger row-level.
TG_NAME	**name**	Nome do trigger disparado
TG_WHEN	**text**	Contém BEFORE ou AFTER
TG_LEVEL	**text**	Contém ROW ou STATEMENT
TG_OP	**text**	Contém INSERT, UPDATE ou DELETE
TG_RELNAME	**name**	Contém o nome da tabela que invocou a trigger

RETORNO DE FUNÇÕES DE TRIGGER

- Uma função de trigger deve retornar NULL ou uma variável que contenha a estrutura da tabela que disparou o trigger.
- Pode ter os seguintes tipos de retorno:

NEW	**RECORD**	Indica que a nova linha inserida será usada para efetuar o comando que disparou a trigger.
OLD	**RECORD**	Indica que a linha apagada será usada para efetuar o comando que disparou a trigger.
NULL	-	Indica que o comando que disparou a trigger deve ser descartado, porém o **código do trigger é executado por completo e a transação não é abortada.**

RETORNO DE FUNÇÕES DE TRIGGER

- CONCLUSÕES
  - Triggers do tipo INSERT e UPDATE terão em seu retorno a variável NEW;
  - Trigger do tipo DELETE terão em seu retorno a variável OLD.
- LEMBRAR QUE:
  - Triggers do tipo DELETE a variável NEW possui valor NULL.
  - Triggers do tipo INSERT a variável OLD possui valor NULL.

EXEMPLO

- Armazenar em uma tabela de auditoria todas as alterações na tabela VENDEDOR, guardando as informações de quem alterou, quando e os novos valores inseridos para o salário.

SOLUÇÃO

```
CREATE OR REPLACE FUNCTION audita_vendedor()
  RETURNS trigger AS
$BODY$
DECLARE
  oldsal double precision;
BEGIN
IF TG_OP = 'UPDATE' THEN
   oldsal := old.salario;
ELSE
   oldsal := NULL;
END IF;
INSERT INTO auditavendedor VALUES (new.idvendedor, oldsal, new.salario, current_timestamp, current_user);
RETURN NEW;
END;
$BODY$
  LANGUAGE 'plpgsql'
```

SOLUÇÃO

```
CREATE TRIGGER trig_audita_vendedor BEFORE
    INSERT OR UPDATE ON vendedor FOR EACH
    ROW EXECUTE PROCEDURE
    audita_vendedor();
```

IGNORANDO A SENTENÇA QUE DISPAROU O TRIGGER

```
CREATE OR REPLACE FUNCTION loginv() RETURNS trigger AS $$
DECLARE
  oldsal double precision;
BEGIN
IF TG_OP = 'UPDATE' THEN
  oldsal := old.salario;
ELSE
  oldsal := NULL;
END IF;
INSERT INTO auditavendedor VALUES (new.idvendedor, oldsal, new.salario, current_timestamp, current_user);
IF current_database() = 'tt' THEN
  RETURN NULL;
ELSE
  RETURN NEW;
END IF;

END;
$$ LANGUAGE 'plpgsql';
```