|  |  |
| --- | --- |
|  | **INSTITUTO FEDERAL DA PARAÍBA - IFPB** **Unidade Acadêmica de Informação e Comunicação**  **CST em Sistemas para Internet** |

**Disciplina: Banco de Dados II**

**Professora: Damires**

**Aluno: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matrícula: \_\_\_\_\_\_\_\_\_\_\_\_\_**

# **Tarefa 9 – Triggers e Funções**

\*\* U*se o BD Filmes*

1. Verifique o seguinte trigger:

Create or Replace Function trocaNome()

Returns trigger as $$

declare msg varchar(40);

Begin

msg = 'Nome '||old.nomeart|| ' mudou para '||new.nomeart;

raise notice 'Foi feito: %',msg;

return null;

End;

$$ LANGUAGE plpgsql;

CREATE TRIGGER veNome AFTER UPDATE

of nomeart ON Artista

FOR EACH ROW

EXECUTE PROCEDURE trocaNome();

select \* from artista;

update artista

set nomeart = 'TROCA'

where nomeart = 'Brad Pitt';

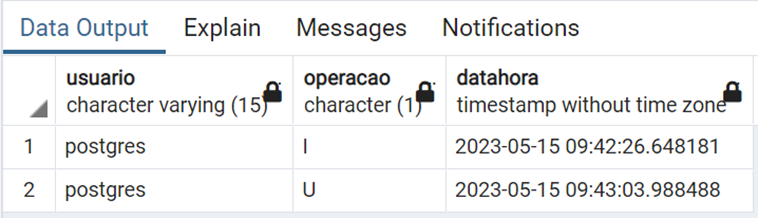
a) O que o trigger faz?

b) Como você realizou o teste do trigger? Como ele foi “disparado”?

c) O que são os objetos “*old*” e “*new*”?

d) Altere de volta o nome do artista.

2. Crie uma tabela *filmeLog* contendo três campos: usuario varchar(20), operacao char(1) e dataHora timestamp. Em seguida, crie um trigger que registre nesse log (filmeLog) as informações de quem fez a operação, o tipo de operação e sua data/hora. Use as funções “user” e “now”. O resultado deve ser como o seguinte:



\*\*Dica: use op = SUBSTR(tg\_op,1,1);

3. Crie uma visão *filmeCateg* que apresente titulo do filme e nome da categoria, ou seja, os títulos dos filmes e suas respectivas categorias (por exemplo, ‘Aventura’). Consulte os dados da visão.

4. Crie um trigger que permita a inserção na tabela filme a partir da view. Se a categoria ainda não estiver cadastrada, faça sua inserção também. Para isso, verifique se a categoria já existe (use a exceção pré-definida *no\_data\_found* para ser lançada quando a categoria não existir).

a. Teste fazendo dois inserts por meio da view.

b. Mostre os resultados na view e nas tabelas.

5. Crie um trigger para efetuar um *Update Cascade* sempre que a tabela estudio tiver sua PK atualizada. A atualização em cascata deve ocorrer na tabela “filha” Filme.

\*\*Dica: Para fins de teste, altere a tabela Filme retirando a constraint de FK: Alter table filme drop constraint fkfilme2estud;

Verifique as alterações. Em seguida, retorne aos valores originais e crie novamente a constraint de FK.

6. Crie uma function armazenada que calcule e retorne a idade do artista. Passe o código do artista como parâmetro.

\*\* Dica: olhe as funções pré-definidas do Postgres.