

AULA 13

TRIGGERS

DEFINIÇÃO

- Recurso que faz o banco de dados ter um pouco mais de “vida própria”.
- Não precisa da dependência constante de uma aplicação ou usuário.
- Tradução: “gatilhos”.

DEFINIÇÃO

- Triggers: disparam a execução de códigos armazenados no servidor, ou outros códigos SQL sem necessidade de uma chamada específica para os mesmos como ocorre com as funções.
- Iniciam a partir de uma condição satisfeita no servidor (antes ou depois de inserção, alteração, exclusão ou horários programados).

VANTAGEM

- Automatização de certas operações quando ocorrem algumas situações no banco de dados.
- Exemplos:
 - Produto esgotando em estoque.
 - Validação de dados.
 - Valor alcançado em orçamento, etc.

VANTAGEM

- Uma aplicação desta forma otimiza os recursos do servidor (não precisa verificar sempre, verifica só quando ocorre alterações)

TRIGGERS

- É possível definir mais de um trigger para determinada condição.
- Neste caso, estes são executados em ordem alfabética na fila de execução do PostgreSQL.
- Neste caso, se o primeiro da fila retornar NULL os demais não são executados.

TRIGGERS

- Apresentam diversas funcionalidades, mas alguns ainda **não** foram resolvidos:
 - Disparo a partir de colunas: executar função a partir da alteração de uma coluna específica
 - Linha antiga e nova: trabalhar ao mesmo tempo com as informações antigas da linha que está sendo alterada com novas informações
 - Comandos SQL: criação de objetos dentro dos blocos de código das funções chamadas por triggers.

TIPOS DE TRIGGERS

- Classificados em duas formas:
 - BEFORE (antes) / AFTER (depois)
 - Comandos: INSERT, UPDATE ou DELETE
- Não pode cadastrar trigger para SELECT.
- Classificado como execução por comando (executado uma única vez) ou execução por linha alterada da tabela (pode ser invocado diversas vezes)

MODOS DE UTILIZAÇÃO

- BEFORE INSERT
- BEFORE UPDATE
- BEFORE DELETE
- AFTER INSERT
- AFTER UPDATE
- AFTER DELETE

CRIANDO UM TRIGGER

```
CREATE TRIGGER nome tipo_trigger eventos ON  
tabela FOR tipo_execução  
EXECUTE PROCEDURE funcao (parametros)
```

EXEMPLO

```
CREATE TABLE total_contas
```

```
(
```

```
  tipo character varying(16),  
  quantidade integer
```

```
)
```

```
INSERT INTO total_contas(tipo, quantidade) VALUES  
('FISICA', 0);
```

```
INSERT INTO total_contas(tipo, quantidade) VALUES  
('JURIDICA', 0);
```

EXEMPLO

```
CREATE TABLE contas  
(  
    numero serial NOT NULL,  
    saldo double precision,  
    tipo character varying(1),  
    CONSTRAINT numero PRIMARY KEY (numero)  
)
```

EXEMPLO

```
CREATE OR REPLACE FUNCTION incrementacontas() RETURNS trigger AS
$BODY$
BEGIN
  IF NEW.tipo = 'F' THEN
    UPDATE total_contas SET quantidade=quantidade + 1 WHERE tipo =
    'FISICA';
  ELSE
    UPDATE total_contas SET quantidade=quantidade + 1 WHERE tipo =
    'JURIDICA';
  END IF;
  RETURN NULL;
END;
$BODY$ LANGUAGE 'plpgsql'
```

EXEMPLO

```
CREATE TRIGGER triggerincrementacontas  
  AFTER INSERT ON contas FOR EACH ROW  
  EXECUTE  
  PROCEDURE incrementacontas();
```

EXEMPLO

```
INSERT INTO contas (numero, saldo, tipo) VALUES (1, 0,  
'F');
```

```
INSERT INTO contas (numero, saldo, tipo) VALUES (2, 0,  
'F');
```

```
INSERT INTO contas (numero, saldo, tipo) VALUES (3, 0,  
'J');
```

```
INSERT INTO contas (numero, saldo, tipo) VALUES (4, 0,  
'J');
```

```
INSERT INTO contas (numero, saldo, tipo) VALUES (5, 0,  
'J');
```

EXEMPLO

```
SELECT * FROM total_contas;
```


ALTERANDO UM TRIGGER

- Para alterar um trigger, é mais prático excluí-lo e criá-lo novamente.
- A única propriedade possível de alteração é seu nome.

`ALTER TRIGGER nome ON tabela RENAME TO novonome;`

EXCLUINDO UM TRIGGER

DROP TRIGGER nome ON tabela opcoes;
[CASCADE, RESTRICT]

HABILITANDO UM TRIGGER

- A habilitação do trigger é automática, mas se por acaso a tabela for configurada para não executar trigger isto pode ser mudado.

```
ALTER TABLE nome_tabela ENABLE TRIGGER  
nome_trigger;
```

- Para desabilitar, utilize DISABLE no lugar de ENABLE.