



Banco de Dados II

Prof. Vinícius Alves Hax



Na aula anterior

- Case
- While



Na aula de hoje

- Triggers

Triggers

- Os triggers (ou “gatilhos” em português) são uma forma de associar a execução automática de funções a comandos SQL
- Exemplo: Para um sistema acadêmico. Quando for feito o INSERT de uma nota no sistema, se houver uma tabela TURMA com o campo MÉDIA, essa média deve ser recalculada após cada nota inserida.

Quando os triggers são acionados?

- Os triggers podem ser acionados ANTES da execução de um SQL ou APÓS
- Quando acionados antes usamos a palavra-chave BEFORE e quando depois usamos a palavra-chave AFTER
- Exemplo de trigger do tipo BEFORE: Alter de executar um DELETE em um administrador do sistema podemos verificar se existe outro administrador cadastrado
- Exemplo de trigger AFTER: Gerar e/ou atualizar dado estatístico em alguma tabela relacionada

Com quais comandos podemos usar os triggers?

- Os triggers podem ser executados ANTES ou DEPOIS dos seguintes comandos: UPDATE, INSERT e DELETE
- Não existem triggers, nas versões atuais do PL/pgSQL para o SELECT
- O Postgres também possui trigger para o comando TRUNCATE (comando que deleta todos os dados de uma tabela)



Peculiaridades do Postgres

- Além de possibilitar triggers para o comando TRUNCATE o Postgres
- Permite o uso de triggers em views
- REQUER que sejam usadas funções como triggers (outros SGBD permitem comandos normais do SQL como ações de um trigger)

Tipos de triggers

- Os triggers podem ser de dois tipos
 - Row-level
 - A função é chamada para cada linha afetada
 - Statement-level
 - A função é chamada para cada comando
- Exemplo:
 - UPDATE alunos SET nota = 6 WHERE nota < 6
 - Trigger row-level: Uma chamada para cada registro afetado
 - Trigger statement-level: Uma única chamada



Criando um trigger

- Para criar um trigger são necessários dois passos
 - Criar a função que será associada ao trigger
 - Associar a função a um determinado evento (INSERT, DELETE, etc)

Criando uma função trigger

- A sintaxe é parecida com a criação de funções normais, exceto que:
 - A função não tem parâmetros
 - O tipo do retorno é o tipo TRIGGER

Variáveis especiais em um trigger

- A função trigger tem acesso a algumas variáveis especiais
- Destacaremos duas, por enquanto, NEW e OLD (disponíveis quando usamos “FOR EACH ROW”)
 - NEW representa o novo registro, que vai ser ou foi inserido/atualizado na tabela
 - OLD representa o registro antigo, que vai ser atualizado/deletado na tabela

Template de função trigger

```
CREATE FUNCTION nome()
```

```
    RETURNS TRIGGER
```

```
    LANGUAGE PLPGSQL
```

```
AS $$
```

```
BEGIN
```

```
    -- corpo do trigger
```

```
END;
```

```
$$
```

Template adição trigger

-- evento pode ser INSERT, DELETE, UPDATE
ou TRUNCATE

```
CREATE TRIGGER nome_trigger
    {BEFORE | AFTER} { evento }
    ON nome_tabela
    [FOR [EACH] { ROW | STATEMENT }]
    EXECUTE PROCEDURE nome_função
```

Exemplo 1

- Ver arquivo trigger1.sql
<https://tinyurl.com/ifsultriggers1>
- Crie os triggers e as funções
- Insira um registro na tabela com o comando abaixo

```
INSERT INTO teacher (id, name, side) VALUES  
(1, 'Palpatine dois', 'Dark');
```

Exemplo 2

- Ver arquivo trigger2.sql
<https://tinyurl.com/ifsultriggers2>
- Crie os triggers e as funções
- Insira um registro na tabela com o comando abaixo

```
INSERT INTO teacher (id, name, side) VALUES  
(1, 'Palpatine dois', 'Dark');
```



Exercícios

(Utilize para ambos as tabelas criadas na lista de exercícios 2)



Exercício 1

- Imagine que nenhum professor pode ter o atributo level menor do que 60. Crie um trigger e uma função correspondente que sempre que for inserido um usuário com nível menor que 60, o nível será alterado para 60.

Exercício 2

- Crie uma tabela auxiliar chamada stats com duas colunas total_power_light e total_power_dark (com valores inicialmente 0)
- Sempre que um novo usuário for inserido o total do lado correspondente ('Light' ou 'Dark') deve ser incrementado
- Também através de triggers padronize o valor do campo de maneira que se o insert for feito usando somente letras minúsculas, a primeira letra deverá sempre ser colocada em maiúsculo



Resumo

- Triggers são muito úteis e bastante utilizados
- Podem ajudar a manter um banco consistente
- Concentram a lógica que precisaria estar dividida em múltiplos locais (ex: site e app móvel)
- Porém podem gerar comportamentos estranhos e bugs difíceis de resolver: use com moderação e documente a solução



Referências

- <https://www.postgresqltutorial.com/postgresql-triggers/>
- <https://www.postgresql.org/docs/current/trigger-definition.html>