SQL (Triggers)

Geomar A. Schreiner
geomarschreiner@gmail.com

Triggers

- Gatilhos(triggers) são objetos acessórios a tabelas e visões que funcionam como “ouvidores” (listeners) de eventos
- O objetivo da criação de triggers é observar ocorrências de inserção, atualização e exclusão de registros ou execução de comandos SQL sobre os objetos aos quais os gatilhos estão vinculados, ANTES ou DEPOIS da sua ocorrência
- É comum que sejam implementados para executar operações que são derivadas, diretamente, de outras operações
- Exemplo: gravar logs de manutenção de dados

Triggers

CREATE [CONSTRAINT] TRIGGER ***name*** { BEFORE | AFTER | INSTEAD OF } { ***event*** [OR ...] }

ON ***table_name***

[FROM ***referenced_table_name***]

[NOT DEFERRABLE | [DEFERRABLE] { INITIALLY IMMEDIATE | INITIALLY DEFERRED }]

[FOR [EACH] { ROW | STATEMENT }]

[WHEN (***condition***)]

EXECUTE PROCEDURE ***function_name*** (***arguments***)

Triggers

- BEFORE, AFTER, INSTEAD OF
  - Permite especificar se a ação do gatilho ocorrerá ANTES do evento que o disparou, DEPOIS do evento ou NO LUGAR do evento
- Event
  - INSERT, UPDATE, DELETE, TRUNCATE
- [FOR [EACH] { ROW | STATEMENT }]
  - A ação do gatilho é repetida para cada registro atingido pelo evento, ou executada apenas uma vez para atender ao evento

Triggers

- [WHEN (condition)]
  - Permite estabelecer uma condição para a execução da ação do trigger
    - exemplo: executar apenas se o registro do funcionário indicar que o mesmo está ativo no cadastro
- EXECUTE PROCEDURE function_name (arguments)
  - Define qual função implementa a ação do trigger
  - Funções que atendem gatilhos são conhecidas como **trigger functions**
  - São **diferentes** de **funções comuns** por retornarem **trigger**

Triggers

- Exemplo

```
CREATE TRIGGER check_update
    BEFORE UPDATE ON accounts
    FOR EACH ROW
    WHEN (OLD.balance IS DISTINCT FROM NEW.balance)
    EXECUTE PROCEDURE check_account_update();
```

Triggers

- Exemplo
  - Médicos com mais de 60 anos não podem trabalhar em andares diferentes do 1.

Triggers

- Exemplo

```
1  CREATE OR REPLACE FUNCTION medico_velho() RETURNS TRIGGER AS $body$
2    DECLARE andar int;
3  BEGIN
4    IF (NEW.nroa IS NULL) THEN
5      return NEW;
6    END IF;
7    EXECUTE 'SELECT andar FROM ambulatorios WHERE nroa = '|| NEW.nroa INTO andar;
8    IF (new.idade >= 60 and andar > 1) THEN
9       RAISE EXCEPTION 'Médico velho demais pra isso!';
10   END IF;
11   RETURN NEW;
12 END;
13 $body$
14 LANGUAGE plpgsql;
```

```
16 CREATE TRIGGER testeTudo BEFORE INSERT OR UPDATE ON medicos
17 FOR EACH ROW EXECUTE PROCEDURE medico_velho();
```

Triggers

- Exemplo 2

```
1  CREATE OR REPLACE FUNCTION registra_log() RETURNS TRIGGER AS $body$
2    DECLARE dados_antigos TEXT; dados_novos TEXT;
3  BEGIN
4    IF (TG_OP = 'UPDATE') THEN
5      dados_antigos := ROW(OLD.*);
6      dados_novos := ROW(NEW.*);
7      INSERT INTO log VALUES (dados_antigos, dados_novos);
8      RETURN NEW;
9    ELSIF (TG_OP = 'DELETE') THEN
10     dados_antigos := ROW(OLD.*);
11     INSERT INTO log VALUES (dados_antigos, DEFAULT);
12     RETURN OLD;
13   ELSIF (TG_OP = 'INSERT') THEN
14     dados_novos := ROW(NEW.*);
15     INSERT INTO log VALUES (DEFAULT, dados_novos);
16     RETURN NEW;
17   END IF;
18 END;
19 $body$
20 LANGUAGE plpgsql;
```

Triggers

- Exemplo 2

```
CREATE TRIGGER log_funcionario
AFTER INSERT OR UPDATE OR DELETE ON funcionario
FOR EACH ROW EXECUTE PROCEDURE registra_log();
```

Exercícios

1) Criar uma trigger que verifique e grave o nome de novos clientes em MAIÚSCULO (função UPPER(varchar));
2) Crie uma nova tabela chamada “log”, com os seguintes atributos: “identificador” (serial), “tabela” (varchar com 50 posições), “operacao” (varchar com 10 posições), “dadosNovos” (texto), “dadosAntigos” (texto);
3) Crie um trigger de log para as tabelas a serem monitoradas via trigger são: “clientes” e ‘locacoes’; A trigger deve fazer as seguintes operações
    a) quando ocorrerem atualizações (UPDATEs) nos registros dessas tabelas, o SGBD deverá inserir registros na tabela “log”, preenchendo seus atributos com o nome da tabela que está sendo modificada, a operação que está sendo executada (“UPDATE”) e o conteúdo anterior e atual dos registros que estão sendo modificados;
    b) quando ocorrerem exclusões (DELETEs) de registros dessas tabelas, o SGBD deverá inserir registros na tabela “log”, preenchendo seus atributos com o nome da tabela cujos registros estão sendo excluídos, a operação que está sendo executada (“DELETE”) e o conteúdo dos registros que estão sendo excluídos.