Bases de Dados II
Introdução à Linguagem PL/SQL

Henrique Madeira, Pedro Bizarro & Bruno Cabral

Exercícios práticos 3 - *Triggers*

Um trigger é um programa em PL/SQL associado a uma tabela e que é armazenado no servidor. A execução do trigger é disparada pela execução de operações de manipulação de dados (DML) sobre a tabela (`INSERT`, `UPDATE` ou `DELETE`) a que o trigger está associado. A execução dos triggers é muito eficiente, pois estes programas são armazenados no servidor já compilados e prontos a serem executados.

Os triggers são utilizados essencialmente com os seguintes objectivos:

- Efectuar verificações de integridade complexas que não possam ser feitas através de métodos declarativos;
- Reforçar mecanismos de segurança na base de dados;
- Geração automática de atributos derivados;
- Permitir registar operações efectuadas na base de dados (i.e., saber quem efectuou determinada operação sobre uma tabela, em que data, os valores alterados, etc.);
- Recolher estatísticas sobre a utilização da base de dados;
- Manutenção de réplicas síncronas.

Vejamos um exemplo de criação de um trigger:

```
CREATE OR REPLACE TRIGGER verifica_stock
  AFTER UPDATE OF quantidade ON Existencias
  WHEN(new.quantidade < new.stock_minimo)
  FOR EACH ROW

DECLARE
  x NUMBER;

BEGIN
  SELECT COUNT(*)
    INTO x
    FROM Encomendas
   WHERE cod_peça = :new.cod_peça;

  IF  x = 0 THEN
    INSERT INTO encomendas
      VALUES(:new.cod_peça, :new.quant_encomendar, sysdate);
  END IF;
END;
```

Se o trigger já existir substitui-o pelo novo código agora definido.

O trigger é executado sempre que é feito um `UPDATE` no atributo `QUANTIDADE` da tabela `EXISTENCIAS` e a execução do trigger acontece depois do `UPDATE` ter tido lugar.

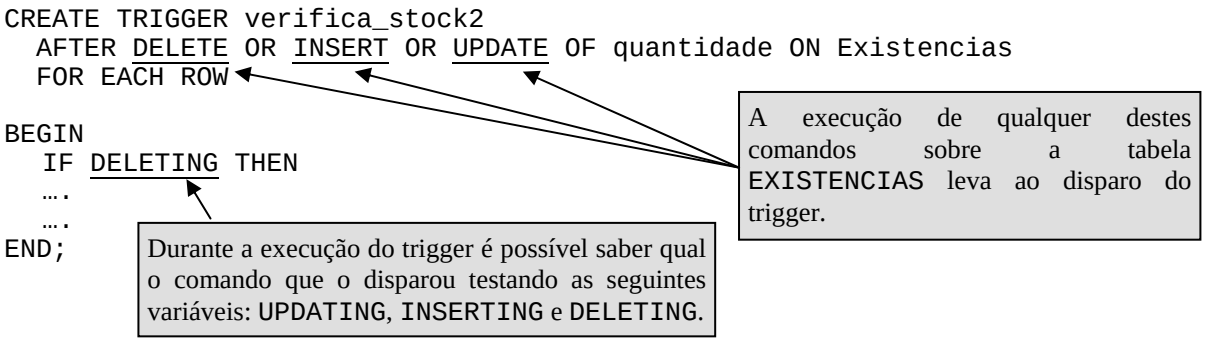
O trigger só é efectivamente executado se esta condição se verificar no momento da sua activação.

Se o comando que actualiza a tabela `EXISTENCIAS` (e que provoca o disparo do trigger) actualizar mais do que uma linha então o trigger é executado para todas as linhas actualizadas.

O corpo do trigger é um programa em PL/SQL. Reparar, no entanto, que é possível ter acesso quer aos valores antigos do registo que está a ser actualizado (`:old.nome_atributo`) quer aos novos valores (`:new.nome_atributo`).

É sempre conveniente testar o trigger antes de o criar (notar que a criação do trigger leva à sua compilação e armazenamento no servidor). Este teste pode ser efectuado executando o código do corpo do trigger como um normal programa PL/SQL anónimo. Convém, no entanto, ter em conta que há coisas que não podem ser testados deste modo. Por exemplo, os valores `:old.nome_atributo` e `:new.nome_atributo` não estão disponíveis quando se executa o corpo do trigger como um normal programa PL/SQL.

Os triggers podem ser eliminados com o comando SQL: **`DROP TRIGGER nome_trigger`**;
É possível definir vários triggers para a mesma tabela, cada um sendo disparado por uma determinado evento e efectuando cada um a sua acção. Contudo, também é possível ter um único trigger a ser disparado por vários eventos. Vejamos um exemplo:



Exercício 1

Fazer um trigger que mantenha o atributo `SalTotal` da tabela `DEP2` permanentemente actualizado. Este atributo (que ainda não existe nesta altura na tabela `DEP2`) regista o montante total gasto mensalmente com a remuneração dos empregados de cada departamento. Deste modo, sempre que o salário, prémios ou departamento de um funcionário são alterados, o `SalTotal` do departamento a que o empregado pertence deve ser alterado em conformidade. A resolução deste problema deve ter os seguintes passos:

a) Criar a tabela `DEP2` (cópia da tabela `DEP`), alterá-la acrescentando o atributo `SalTotal` do tipo `NUMBER`. Depois desta alteração a estrutura da tabela `DEP2` deve ficar com o seguinte aspecto:

```
SQLWKS> DESC DEP2;
Column Name                    Null?    Type
------------------------------ -------- ----
NDEP                                    NUMBER(2)
NOME                                    VARCHAR2(15)
LOCAL                                   VARCHAR2(15)
SALTOTAL                                NUMBER
```

b) Fazer um programa em PL/SQL que preencha correctamente o campo `SalTotal` da tabela `DEP2`. Este programa corresponde à inicialização dos dados.

Solução

c) Fazer o trigger pedido (sugestão: faça primeiro o trigger sem considerar os prémios).

Solução