
Banco de Dados II

Triggers

Profa.: Márcia Sampaio Lima

EST - UEA

Triggers

- Conceito:

- TRIGGER (gatilho) é um objeto de BD, associado a uma tabela, disparado em resposta a um evento em particular.

- Exemplo de eventos:

- INSERT, REPLACE, DELETE ou UPDATE
 - DML (Data Manipulation Language)
-

Triggers

- Pode-se definir vários TRIGGERS em uma base de dados.
 - Cada Trigger deve ser associada a uma tabela e a um evento que irá dispará-la.
-

Triggers

- Quando os triggers são executados?
 - Podem ser disparados nos seguintes MOMENTOS :
 - antes (BEFORE) ou
 - depois (AFTER) do evento.
-

Trigger

**Para cada tupla (tabela, evento, momento),
podemos definir apenas UMA Trigger.**

Triggers

- Os principais pontos positivos sobre os **triggers** são:
 - ❑ Parte do processamento que seria executado na aplicação passa para o banco, poupando recursos da máquina cliente.
 - ❑ Facilita a manutenção, sem que seja necessário alterar o código fonte da aplicação.
-

Triggers

- Já contra sua utilização existem as seguintes considerações:
 - ❑ Alguém que tenha acesso não autorizado ao banco de dados poderá visualizar e alterar o processamento realizado pelos gatilhos.
 - ❑ Requer maior conhecimento de manipulação do banco de dados (SQL) para realizar as operações internamente.
-

Triggers

- Criando Trigger no MySQL:

```
CREATE TRIGGER nome momento evento  
ON tabela  
FOR EACH ROW  
  BEGIN  
    /*corpo do código*/  
  END
```

Triggers

■ Onde:

- ❑ nome: nome do gatilho.
 - ❑ momento: quando o gatilho será executado.
Valores válidos são BEFORE (antes) e AFTER (depois).
 - ❑ evento: evento que vai disparar o gatilho: **INSERT, UPDATE e DELETE.**
 - ❑ tabela: nome da tabela a qual o gatilho está associado.
-

Triggers

- Não é possível criar mais de um trigger para o mesmo evento e momento de execução na mesma tabela.
 - Por exemplo, não se pode criar dois gatilhos **AFTER INSERT** na mesma tabela.
-

Triggers

- Como os *triggers*, são executados em conjunto com operações de inclusão e exclusão, é necessário poder acessar os registros que estão sendo incluídos ou removidos. Isso pode ser feito através das palavras NEW e OLD.
 - NEW dá acesso ao novo registro.
 - OLD dá acesso ao registro que está sendo removido
-

Triggers

■ Exemplo: Exercício Prático 1

- ❑ Clientes compram livros.
 - ❑ Clientes(idCli, nome, endereco)
 - ❑ Livros (codigo, nome, ano_pub, qtdeDispCompra)
 - ❑ Fazer MER
-

Triggers

■ Criar Tabela Livros:

```
CREATE TABLE Livros(  
    codigo    INT(11) PRIMARY KEY,  
    nome      VARCHAR(30),  
    ano_pub  int(11),  
    qtdeEstoque  INT NOT NULL DEFAULT 0  
);
```

Triggers

■ Povoar Tabela:

```
INSERT INTO Livros VALUES (1, 'livro BD',2015, 4);  
INSERT INTO Livros VALUES (2, 'livro de SO',2016, 12);  
INSERT INTO Livros VALUES (3, 'Python', 2017,35);
```

Triggers

- Cria tabela ItemCompra

```
CREATE TABLE ItensCompra(  
    notaFiscal          INT PRIMARY KEY,  
    livro      INT,  
    Quantidade  INT  
);
```

Triggers

```
ALTER TABLE itenscompra  
ADD CONSTRAINT fk_itenscompra_livro FOREIGN KEY (livro)  
REFERENCES livros (codigo);
```

Triggers

■ Problema:

- ❑ Ao inserir e remover registro da tabela ItensCompra, o estoque do produto (na tabela livros) deve ser alterado.

■ Solução:

- ❑ Serão criados dois triggers:
 - *AFTER INSERT* para dar baixa no estoque
 - *AFTER DELETE* para fazer a devolução da quantidade do produto.
-

Criando Trigger - Exemplo 1

```
DELIMITER $
```

```
CREATE TRIGGER Tgr_ItensCompra_Insert AFTER INSERT  
ON ItensCompra
```

```
FOR EACH ROW  
BEGIN
```

```
    UPDATE livros SET qtdeEstoque= qtdeEstoque -  
    NEW.Quantidade WHERE codigo = NEW.livro;
```

```
END$
```

```
DELIMITER ;
```

Triggers

- O NEW foi usado para obter as informações da linha que está sendo inserida na tabela.
 - TESTANDO
INSERT INTO itensCompra VALUES (13, 3,2);
-

Triggers

- EXCLUIR TRIGGER
 - ❑ DROP TRIGGER IF EXISTS
Tgr_ItensCompra_Insert;



Criando Trigger - Exemplo 2

```
DELIMITER $
```

```
CREATE TRIGGER Tgr_ItensCompra_Delete AFTER DELETE  
ON ItensCompra
```

```
FOR EACH ROW  
BEGIN
```

```
    UPDATE livros SET qtdeEstoque = qtdeEstoque +  
    OLD.Quantidade WHERE codigo= OLD.livro;
```

```
END$
```

```
DELIMITER ;
```

Triggers

■ TESTANDO

- ❑ DELETE FROM `itenscompra` WHERE
`notaFiscal`=13;

Triggers

- CRIANDO OUTRA TRIGGER PARA O MESMO EVENTO, MOMENTO E TABELA:

```
DELIMITER $
```

```
CREATE TRIGGER Tgr2_ItensCompra_Delete AFTER DELETE  
ON ItensCompra
```

```
FOR EACH ROW  
BEGIN
```

```
    UPDATE livros SET qtdeCompra = qtdeCompra +  
    OLD.Quantidade WHERE idLivro = OLD.livro;
```

```
END$
```

```
DELIMITER ;
```

Triggers

■ Erro:

Mensagens do MySQL :

#1235 - This version of MySQL doesn't yet support 'multiple triggers with the same action time and event for one table'

Triggers

- VISUALIZANDO AS TRIGGERS EXISTENTES:

show triggers;

Triggers

- Em ambientes reais, triggers podem ser utilizados para operações mais complexas.
 - Exemplo:
 - Antes de vender um item, verificar se há estoque disponível e só então permitir a saída do produto.
-