

# Banco de Dados II

## Parte 8 – Trigg

Professor Eduardo

# Introdução

- **Trigger** (também chamado de **gatilho**) e é um tipo de procedimento armazenado que é disparado somente quando o evento pré-definido é acionado facilitando o uso de regras de integridade.

- No momento da sua criação cada gatilho é associado a um evento de manipulação de dados (insert, update ou delete) em um objeto do BD e só é acionado diante da ocorrência deste evento determinado.



# Sintaxe

```
CREATE TRIGGER nome_trigger  
momento_execução  
evento_disparador  
ON nome_tabela  
FOR EACH ROW  
Comando_SQL;
```

**Onde:**

**Momento\_execução: BEFORE(antes)/AFTER(depois)**

**Evento\_disparador: INSERT/UPDATE/DELETE**

**Se for um grupo de comandos SQL eles devem ser delimitados por BEGIN ... END**

# Exemplo

Antes de inserir uma venda a quantidade vendida é retirada do estoque.

```
INSERT INTO VENDAS (ID_VENDA, ID_PRODUTO, DATA_VENDA,  
QTD_VENDIDA, ID_CLIENTE) VALUES (123, 10, '2011-01-01', 38, 200);  
  
CREATE TRIGGER GATILHO_VENDAS BEFORE INSERT ON VENDAS  
FOR EACH ROW  
UPDATE ESTOQUES  
SET QTD_ESTOCADA = QTD_ESTOCADA-NEW.QTD_VENDIDA  
WHERE IDPRODUTO = NEW.IDPRODUTO
```

A quantidade vendida que foi informada no comando INSERT disparador o gatilho é acessada a partir da tabela NEW

# Tabelas Auxiliares

- Em caso de INSERT:
  - O operador NEW.nome\_coluna, nos permite verificar o valor enviado para ser inserido em uma coluna de uma tabela.
  - OLD.nome\_coluna não está disponível.
- Em caso de DELETE:
  - O operador OLD.nome\_coluna nos permite verificar o valor excluído ou a ser excluído.
  - NEW.nome\_coluna não está disponível.
- Em caso de UPDATE:
  - Tanto OLD.nome\_coluna quanto NEW.nome\_coluna estão disponíveis, antes (BEFORE) ou depois (AFTER) da atualização de uma linha.

# Exemplo

Antes de inserir uma venda a quantidade vendida é retirada do estoque.

```
DELIMITER $$  
CREATE TRIGGER GATILHO_ESTOQUES  
AFTER UPDATE ON ESTOQUES  
FOR EACH ROW  
BEGIN  
    DECLARE QTD_FABRICA INT DEFAULT 0;  
    IF (NEW.QTD_ESTOCADA < NEW.QTD_MINIMA)  
        THEN  
        SET QTD_FABRICA = NEW.QTD_MINIMA - NEW.QTD_ESTOCADA;  
        INSERT INTO FABRICACAO_PREVISTA (IDPRODUTO, DATA_SOLICITACAO, QUANTIDADE  
            SITUACAO) VALUES (NEW.IDPRODUTO, NOW(), QTD_FABRICA, 'PENDENTE');  
        END IF;  
    END;  
$$  
DELIMITER ;
```

## OBS:

A função **CURDATE()** recupera a data atual.  
A função **NOW()** recupera a data e a hora atual.

Depois de alterar o estoque, se a quantidade estocada for menor do que a quantidade mínima, insere um registro na tabela de previsão de fabricação com a quantidade a ser fabricada, sendo a mínima menos a estocada.

# Sintaxe

Para eliminar um trigger:

**DROP TRIGGER IF EXISTS GATILHO\_VENDAS;**

# Exemplo

```
DELIMITER $$
CREATE TRIGGER GATILHO_VENDAS BEFORE INSERT ON VENDAS
FOR EACH ROW
BEGIN
    DECLARE QTD_AUX INT;
    SELECT QTD_ESTOCADA INTO QTD_AUX FROM ESTOQUES
        WHERE IDPRODUTO=NEW.IDPRODUTO;
    IF (QTD_AUX < NEW.QTD_VENDIDA)
        THEN SET NEW.QTD_VENDIDA=QTD_AUX;
    END IF;
    UPDATE ESTOQUES
        SET QTD_ESTOCADA = QTD_AUX-NEW.QTD_VENDIDA
        WHERE IDPRODUTO = NEW.IDPRODUTO;
END;
$$
DELIMITER ;
```

Antes de ins  
venda, procu  
estoque o  
vendido, se q  
do estoque f  
que quantidade  
altere a q  
vendida para  
máximo em e  
Subtrai em e  
quantidade vend



# Gatilhos DDL

- Os gatilhos DDL são disparados em resposta a diversos eventos DDL (linguagem de definição de dados).
  - Esses eventos correspondem principalmente a instruções SQL que criam, alteram ou eliminam objetos no banco de dados (CREATE, ALTER, DROP,...) ou configuram opções de segurança (GRANT, DENY or REVOKE).
  - Determinados procedimentos armazenados do sistema que executam operações do tipo DDL também podem disparar gatilhos DDL.
- Use gatilhos DDL quando quiser:
  - Evitar determinadas alterações em seu esquema de banco de dados.
  - Ocorrer algo no banco de dados em resposta a uma alteração em seu esquema de banco de dados.
  - Registrar alterações ou eventos no esquema de banco de dados.

# Exemplo

```
CREATE TRIGGER Aluno  
ON DATABASE  
FOR DROP_TABLE, ALTER_TABLE  
AS  
PRINT 'A estrutura desta tabela não pode ser modificada'  
ROLLBACK ;
```

Exemplo usando a sintaxe  
Transact-SQL/MS SQL Server

# Gatilhos “Instead of”

- Os gatilhos **INSTEAD OF** substituem a execução do comando **DML** original (**INSERT**, **UPDATE**, **DELETE**) pela execução dos comandos definidos no gatilho.
- Um gatilho **INSTEAD OF** pode realizar ações como:
  - Ignorar partes de um conjunto de dados.
  - Não processar uma parte de um conjunto de dados e registrar as linhas com problemas.
  - Realizar uma ação alternativa quando um comando é executado.
  - Realizar uma ação alternativa quando uma condição de erro é encontrada.

# Exemplo

```
CREATE TRIGGER InsteadTriggerEstados on TabelaEstados  
INSTEAD OF INSERT  
AS  
BEGIN  
/* Impede que sejam inseridos novos estados. */  
/* Em vez de inserir o estado grava mensagem de auditoria */  
INSERT INTO TabelaAuditoria (DataHoraEvento, Mensagem)  
SELECT GETDATE(), 'Tentativa de inserir novo estado'  
END;
```

Exemplo usando a sintaxe  
Transact-SQL/MS SQL Server

# Referências Bibliográficas

- ELSMARI, Ramez & NAVATHE, Shamkant B. Sistema de Banco de Dados. Rio de Janeiro: Editora Addison-Wesley. 4ª. edição, 2005, 744p.