

Microsoft SQL Server 2016 (T-SQL)

Aula 18

Professor Anderson Henrique

Assuntos tratados nessa aula:

01 – Triggers – Definição e utilização de trigger

Professor Anderson Henrique

01 – Definição e Tipos Instead Of e After

São gatinhos dentro do nosso banco de dados, são rotinas que criamos para automatizar alguma parte do nosso código. Podemos utilizar as triggers para:

Criar restrições no nosso banco de dados

Para criar tabelas de logs

Inserir registros, Excluir registros e outros

Uma trigger é um gatilho que é disparado quando algum evento acontece no nosso banco de dados

-- vamos criar o seguinte banco de dados

CREATE DATABASE aula_trigger

ON PRIMARY (

NAME = aula_trigger,

FILENAME = 'C:\sql\db_aula_trigger.MDF',

SIZE = 16MB,

MAXSIZE = 32MB,

FILEGROWTH=10%

);

GO

--vamos criar a tabela de saldos

```
CREATE TABLE tbSaldos(  
  PRODUTO VARCHAR(10),  
  SALDO_INICIAL NUMERIC(10),  
  SALDO_FINAL NUMERIC(10),  
  DATA_ULT_MOV DATETIME  
);  
GO
```

Essa tabela vai controlar o saldo do produto após a movimentação em uma tabela de movimento

Vamos inserir dados na tabela de saldos

--vamos inserir dados na tabela

```
INSERT INTO tbSaldos (PRODUTO, SALDO_INICIAL, SALDO_FINAL,  
DATA_ULT_MOV)
```

```
VALUES ('Produto A', 0, 100, GETDATE());
```

```
GO
```

Vamos criar a tabela de vendas, onde iremos registrar o movimento de saída desses produtos

--vamos criar a tabela de vendas

```
CREATE TABLE tbVendas(  
ID_VENDAS
```

```
INT,
```

```
PRODUTO
```

```
QUANTIDADE
```

```
DATATIME
```

```
);
```

```
GO
```

Vamos criar uma sequence, serve para sequenciar o histórico de vendas

--vamos criar uma sequence definir sequencia para histórico de Vendas

```
CREATE SEQUENCE seq_tbVendas  
AS NUMERIC  
START WITH 1  
INCREMENT BY 1
```

E, por último criamos uma tabela de histórico de vendas

--vamos criar a tabela de histórico de vendas

```
CREATE TABLE tbHistoricoVendas(  
PRODUTO VARCHAR(10),  
QUANTIDADE INT,  
DATA_VENDA DATETIME  
);  
GO
```

Vamos criar um trigger de ajuste de saldo, onde estrutura funcionará da seguinte forma:

Toda vez que eu fizer uma venda na tabela de vendas, eu quero que automaticamente vá na tabela saldos, pegue o produto e o saldo final e subtraia o saldo final do total que foi vendido e retorna o saldo final atualizado.

Esse trigger será disparada na tabela de vendas, todas vez que fizer um insert nessa tabela

--criando a trigger

```
CREATE TRIGGER trg_AjustaSaldo
```

```
ON tbVendas
```

```
FOR INSERT
```

```
AS
```

```
BEGIN
```

```
DECLARE @QUANTIDADEINT,
```

```
    @DATADATETIME,
```

```
@PRODUTOVARCHAR(10)
```



```
SELECT @DATA = DATA, @QUANTIDADE = QUANTIDADE, @PRODUTO =  
PRODUTO FROM INSERTED
```

```
UPDATE tbSalDOS
```

```
SET SALDO_FINAL = SALDO_FINAL - @QUANTIDADE,
```

```
DATA_ULT_MOV = @DATA
```

```
WHERE PRODUTO = @PRODUTO;
```

```
INSERT INTO tbHistoricoVendas (PRODUTO, QUANTIDADE,  
DATA_VENDA)
```

```
VALUES (@PRODUTO, @QUANTIDADE, @DATA)
```

```
END
```

```
GO
```

Agora vamos testar a nossa trigger (gatilho) inserindo uma venda

-- vamos inserir uma venda para disparar o gatilho

```
INSERT INTO tbVendas (ID_VENDAS, PRODUTO, QUANTIDADE, DATA)
VALUES (NEXT VALUE FOR seq_tbVendas, 'Produto A', 2,
GETDATE());
```

2 linhas serão afetadas

Vamos consultar as tabelas para ver o resultado

```
SELECT * FROM tbVendas;
```

```
SELECT * FROM tbSalDOS;
```

```
SELECT * FROM tbHistoricoVendas;
```

Dúvidas?

Professor Anderson Henrique

