

TRIGGERS SQL

Este documento foi elaborado pelo grupo 6 para atender o escopo do projeto sala de aula invertida que tem por objetivo incitar aos alunos envolvidos no projeto saírem de seus respectivos papéis de aluno e verificarem como montar e explicar uma aula. Dessa forma, a aula desenvolvida foi sobre o uso dos triggers no banco de dados SQL SERVER ao qual abordaremos a seguir.

Afinal, o que são Triggers?

Trigger como o próprio nome indica trata-se de um termo que traduzido do inglês significa gatilho, assim só é possível utilizá-lo mediante uma ação que leve ao disparo do gatilho. Podemos dizer que se trata de uma função que é acionada de forma automática após um comando ser executado.

Os *triggers* podem ser criados para comandos DDL e DML. Porém, como não temos muito tempo, iremos explicar nesta aula apenas os *triggers* para comandos DML.

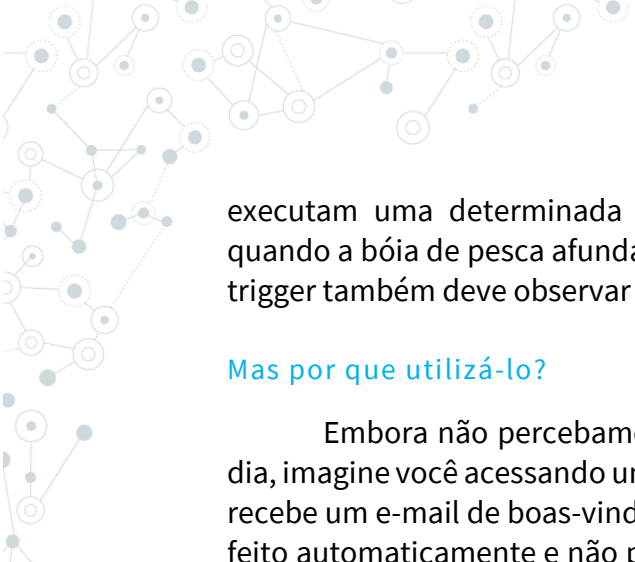
Para entendermos melhor...

Para entender melhor como funcionam os Triggers, podemos fazer uma analogia bem simples com o ato de pescar.



Para que o pescador possa pescar um peixe, ele joga o anzol na água e espera um sinal de que o peixe mordeu o anzol para então poder puxar a vara e capturar o peixe. Esse sinal pode vir da boia, da linha ou até mesmo da vara de pesca.

De maneira semelhante, podemos dizer que os Triggers são como os pescadores. Eles esperam um sinal para serem disparados, e quando isso acontece



executam uma determinada ação. No exemplo do pescador, ele deve observar quando a bóia de pesca afunda para puxar a vara e pegar o peixe, da mesma forma o trigger também deve observar uma determinada ação para que seja disparado.

Mas por que utilizá-lo?

Embora não percebamos o trigger é extremamente utilizado em nosso dia a dia, imagine você acessando um site e cadastrando o seu e-mail, após o cadastro você recebe um e-mail de boas-vindas e com diversas informações do site, mas isso é algo feito automaticamente e não por alguém que está programando? E o que seria isso? Bem é o uso do trigger após a inserção do e-mail o sistema dispara um gatilho onde deve enviar um e-mail de boas-vindas para a pessoa cadastrada. Toda vez que eu preciso inserir, deletar ou modificar alguma informação em diversos locais de forma automática eu utilizo o trigger.

Outro exemplo de uso do trigger

Imagine que temos um banco de dados com 500 filmes e precisamos inserir um valor de 5 minutos a mais na duração dos filmes referentes a uma nova leva de trailers, embora alguns comandos façam isso por nós, isso seria um pouco complicado e desgastante por ter que inserir um por um, pensando nisso, utiliza-se os Triggers, onde podemos criar um trigger que após um comando de INSERT ou UPDATE leva a um acionamento do gatilho, que resultará na atualização dos 500 filmes de uma única só vez.

Tipos de Triggers

Existem 3 tipos principais de triggers, embora com sintaxe semelhante cada um deles é colocado de acordo com o tempo de ação que será executado, ou seja, o seu modo de disparo. Voltando ao exemplo do e-mail, como ele só manda o e-mail após a inclusão do registro no banco de dados trata-se de um trigger **AFTER**, pois necessita que uma ação seja concluída antes de executá-lo. Já para o exemplo do pescador como ele reage assim que o peixe pega a isca temos um exemplo de **FOR** onde a execução do trigger é conjunto com a ação e não após concluí-la. Contudo, essa diferença é vista apenas em casos muito específicos e em geral utilizamos tanto AFTER quanto FOR como semelhantes. Por fim temos o **INSTEAD OF** que diferentemente dos demais é executado no lugar da ação que o geraria.

Sintaxe

Conforme mencionado, a sintaxe geral dos triggers é semelhante, o que muda é o seu modo de disparo, na figura a seguir analisamos a sintaxe geral para a criação dos triggers:





SINTAXE GERAL

```
CREATE TRIGGER nome_trigger  
ON nome_tabela  
[WITH ENCRYPTION]  
AFTER/FOR | INSTEAD OF [INSERT, UPDATE, DELETE]  
AS  
Código do Trigger
```

Fonte: Autoria própria

Conforme visto, para a criação do trigger primeiro devemos definir os seus parâmetros como nome do trigger e em qual tabela ele deverá ser utilizado, além disso já na etapa de criação do trigger podemos definir se iremos utilizá-lo com algum tipo de criptografia, caso não seja necessário basta não citar essa parte no código. Na sequência devemos selecionar o modo de disparo bem como os parâmetros que servirão para o acionamento do gatilho, lembrando que eles podem ser: INSERT, UPDATE ou DELETE e até mesmo uma combinação desses parâmetros. Após todas essas informações é que adicionamos o que o trigger deverá executar após o seu acionamento.

Sintaxe AFTER + Exemplo 'PRINT'


Conforme vemos no exemplo abaixo a mudança foi a seleção do modo de disparo, neste caso definimos como AFTER e selecionamos que ele irá ser executado toda vez que for executado um comando INSERT.

SINTAXE AFTER

```
CREATE TRIGGER trg_mensagem_after  
ON Produtos  
AFTER INSERT  
AS  
PRINT 'Produto adicionado com sucesso!'
```

Fonte: Autoria própria

Para entender melhor como esse código funciona iremos criar um banco de dados de produtos onde a cada vez que adicionarmos um novo produto aparecerá a mensagem de “Produto adicionado com sucesso”.



```
CREATE DATABASE Produtos;
USE Produtos;
CREATE TABLE Produtos
(
    idProduto INT PRIMARY KEY IDENTITY
    ,Nome VARCHAR(100)
    ,Preco FLOAT
);
```

Fonte: Autoria própria

Após criar a tabela produtos iremos adicionar um produto:

```
-- INSERIR DADOS NA TABELA PRODUTOS
INSERT INTO Produtos (Nome, Preco)
VALUES ('Refrigerante', 7.50);
```

Mensagens

(1 linha afetada)

Horário de conclusão: 2021-03-07T11:52:54.3608523-03:00

Fonte: Autoria própria

Perceba que após executar e adicionar o produto no campo mensagens apenas temos a informação de que uma linha foi afetada, assim se quisermos confirmar se os dados foram criados com sucesso podemos utilizar um trigger com modo de disparo para que ele adicione uma mensagem toda vez que um novo produto foi adicionado, desta forma criamos o trigger a seguir:

```
-- TRIGGER AFTER
CREATE TRIGGER trg_mensagem_after -- Nome do trigger
ON Produtos -- Atribuição à tabela produtos
AFTER INSERT -- Após a inserção de um novo registro
AS
PRINT 'Produto inserido com sucesso!'; -- Código do trigger
```

Fonte: Autoria própria

Após a criação e execução do trigger iremos analisar o seu funcionamento adicionando um novo produto:

```
INSERT INTO Produtos (Nome, Preco)
VALUES ('Bolacha', 2.50);
```

Mensagens

Produto inserido com sucesso!

(1 linha afetada)

Horário de conclusão: 2021-03-07T11:57:07.5389912-03:00

Fonte: Autoria própria

Perceba que agora assim que o trigger recebeu o gatilho e percebeu que todo o comando insert estava executado mostrou no campo mensagens a mensagem “Produto inserido com sucesso!”, dessa forma o trigger pode incluir mensagens de aviso de acordo com o que o usuário necessitar.

Sintaxe INSTEAD OF + Exemplo ‘PRINT’

Podemos fazer um exemplo semelhante só que com o uso do trigger com modo de disparo INSTEAD OF, dessa forma ao invés dele realizar a operação que ativa o gatilho do trigger ele sobrescreve as informações para aquela que está presente no trigger.

Para isso, primeiro verificaremos a sintaxe do trigger INSTEAD OF:

SINTAXE INSTEAD OF

```
CREATE TRIGGER trg_mensagem_instead_of
ON Produtos
INSTEAD OF INSERT
AS
PRINT 'Hoje não'
```

Fonte: Autoria própria

Conforme vemos a mudança foi no modo de disparo, contudo para que possamos utilizá-lo primeiro precisamos desabilitar o trigger presente na tabela produtos a fim de não gerar conflitos, para isso utilizaremos o comando a seguir:

```
SELECT [name], is_disabled FROM sys.triggers
WHERE is_disabled = 1 or is_disabled = 0;
```

Fonte: Autoria própria

Nesse caso verificamos todos os triggers ativos e desativados do sistema, assim teremos o resultado no campo mensagens:

	name	is_disabled
1	trg_mensagem_after	0

Fonte: Autoria própria

A pergunta lógica que é realizada tem como resultado true = 1 ou false = 0, e é se o trigger está desabilitado, conforme vemos a resposta é 0=false nos indicando que ele está habilitado. Para desabilitar utilizaremos o comando a seguir:

```
--DESABILITAR TRIGGER
ALTER TABLE Produtos
DISABLE TRIGGER trg_mensagem_after;
```

Fonte: Autoria própria

Voltando a verificar vemos que o resultado da pergunta se está habilitado ou não mudou de valor:

	Resultados	Mensagens
	name	is_disabled
1	trg_mensagem_after	1

Fonte: Autoria própria

Indicando que agora o meu trigger está desabilitado, dessa forma prosseguiremos com a criação do trigger de modo de disparo INSTEAD OF:

```
--TRIGGER INSTEAD OF
CREATE TRIGGER trg_mensagem_instead_of -- Nome do trigger
ON Produtos -- Atribuição à tabela produtos
INSTEAD OF INSERT -- Ao invés de inserir o registro, será executado o código do trigger
AS
PRINT 'Hoje não'; -- Código do trigger

DROP TRIGGER trg_mensagem_after;
DROP TRIGGER trg_mensagem_instead_of;
```

Fonte: Autoria própria

Após a inserção dos parâmetros corretos iremos executar o trigger para criá-lo e habilitá-lo e depois disso iremos testar a inserção de um novo valor na tabela produtos.

```
-- INSERIR DADOS NA TABELA PRODUTOS
INSERT INTO Produtos (Nome, Preço)
VALUES ('Biscoito', 2.50);
```

Mensagens

Hoje não

(1 linha afetada)

```
-- EXIBIR DADOS DA TABELA PRODUTOS
SELECT Produtos.Nome, Produtos.Preço FROM Produtos;
```

	Resultados	Mensagens
	Nome	Preço
1	Refrigerante	7,5
2	Bolacha	2,5

Fonte: Autoria própria

Conforme vemos, ao acionar o gatilho insert, ao invés de fazer a modificação sugerida pelo usuário, foi informado a mensagem “Hoje não” para indicar que essa ação não pode ser realizada, após verificado na tabela produtos vemos que realmente o valor definido no insert não foi inserido, mas foi substituído pelo que constava no valor do trigger.

Por fim, ao acessarmos para verificar os triggers existentes no sistema, vemos que consta agora dois triggers um habilitado e o outro desabilitado.

	Resultados	Mensagens
	name	is_disabled
1	trg_mensagem_after	1
2	trg_mensagem_instead_of	0



Fonte: Autoria própria

Para mais informações...


Para acessar os diversos exemplos citados no texto e alguns exercícios mais elaborados, acesse o Github a seguir:

<https://github.com/vinixiii/Aula-Invertida_Triggers-SQL>

Github com todos os materiais utilizados nesse projeto.

Nele você encontra todo o material de apoio desenvolvido neste projeto incluindo uma apresentação em powerpoint que demonstra o que foi apresentado no projeto sala de aula invertida.

Além disso, há um exercício comentado sobre Triggers Aninhados que se trata do uso de triggers com modo de disparo AFTER sendo utilizados em conjunto.





Referências

DELVA, Philemon. Trigger em SQL. 2017- Alura. Disponível em:
<https://www.alura.com.br/artigos/trigger-em-sql?gclid=CjwKCAiAp4KCBhB6EiwAxRxbpHlxYuOHAq6-vHVbF1Cl6ui0Ae98ML7liw7rWbhw_hBXi3R7VZjhhoCAWAQAvD_BwE> Acesso em 04 de Março de 2021.

DOCUMENTAÇÃO. CREATE TRIGGER (Transact-SQL). 2019. Disponível em:<<https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver15>> Acesso em 04 de Março de 2021.

RODRIGUES, Joel. Triggers no SQL Server: teoria e prática aplicada em uma situação real. 2013 - DevMedia. Disponível em:
<<https://www.devmedia.com.br/triggers-no-sql-server-teoria-e-pratica-aplicada-em-uma-situacao-real/28194>> Acesso em 04 de Março de 2021.

