

O que são Triggers:

O termo “Trigger” (do inglês, gatilho) se refere a uma função disparada dentro do banco de dados com propriedades de: Alterar, atualizar e deletar dados (ALTER, UPDATE e DELETE). Os gatilhos estão intimamente relacionados às tabelas, por tanto, um trigger pode ser acionado quando estamos atualizando um dado em uma tabela específica (com específica quero dizer a tabela onde o trigger foi colocado), quando estamos inserindo um novo dado ou até mesmo deletando.

Portanto, o Trigger nada mais é do que um “Acionador” de uma função específica que você quer deixar habilitada em uma tabela!

Classes de Triggers no SQL:

Triggers DDL (Data Definition Language): Essa classe de Triggers é acionada em eventos que alteram a estrutura (CREATE), modificar(ALTER) ou soltar(DROP)) ou em determinados eventos relacionados ao servidor, como alterações de segurança ou atualização de eventos estatísticos(eventos estáticos são:).

Triggers DML (Data Modification Language): Esta é a classe de Triggers mais usada. Nesse caso, o evento de disparo é uma declaração de modificação de dados; poderia ser uma instrução de inserção, atualização ou exclusão em uma tabela ou em uma exibição.

Tipos de Triggers: Uma trigger pode ser de 3 tipos diferentes, sendo eles: FOR, AFTER e o INSTEAD OF.

FOR: é o valor padrão e indica que a trigger será executada JUNTO com o comando de INSERT, DELETE ou UPDATE que a disparou.

AFTER: faz com que a trigger seja executada APÓS a ação que a gerou ser concluída.

INSTEAD OF: faz com que o trigger seja executado NO LUGAR da ação que o gerou.

Orientações básicas: - Sintaxe

Como criar um Trigger: Para criar um Trigger, é utilizado o seguinte comando no SQL:

```
1 CREATE TRIGGER [NOME DO TRIGGER]
2 ON [NOME DA TABELA]
3 [FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]
4 AS
5     --CORPO DO TRIGGER
```

Como HABILITAR OU DESABILITAR um Trigger: Para HABILITAR ou DESABILITAR triggers em uma tabela de uma forma geral, podemos utilizar o comando no DDL, o ALTER TABLE.

```
ALTER TABLE nome_tabela  
ENABLE|DISABLE TRIGGER nome_trigger
```

Como alterar o conteúdo de um Trigger:

O comando ALTER suporta a alteração de vários objetos criados no banco de dados. Podemos facilmente chamar para alterar o conteúdo de TRIGGER, alterar a palavra CREATE para ALTER e, em seguida, executar o comando.

Nota: Para definir um gatilho, também podemos usar o comando COM ENCRYPTION, mas se você não se lembra do que foi escrito em TRIGGER, você não pode alterá-lo.

Exemplo:

PRINT

Como apagar um Trigger:

Se você deseja excluir TRIGGER do banco de dados, use DROP TRIGGER mais o nome do TRIGGER que deseja excluir;

Exemplo prático:

PRINT

As vantagens de se usar o Trigger:

Exemplos práticos:

Gerar alguns valores de coluna derivados automaticamente:

- Descrição: Temos um banco com tabelas de produtos e vendas, quando um item é vendido e isso é referenciado na tabela de “produtos” poderíamos ter um gatilho de “vendido” nessa tabela, com base nisso, quando o trigger da tabela for acionado ele atualiza os dados na tabela de “Vendas do mês” automaticamente.

Exemplo prático:

PRINT

Aplicar a integridade referencial:

O que é integridade referencial:

- Em resumo, **integridade referencial** é um conceito de banco de dados que garante que todos os relacionamentos propostos entre tabelas no modelo de entidade-relacionamento (ER) serão respeitados dando a certeza que os dados de um banco de dados estarão íntegros, sendo mais específico, **integridade referencial** é um conceito relacionado à chaves estrangeiras.
Este conceito diz que o valor que é chave estrangeira em uma tabela destino, deve ser chave primária de alguma tabela de origem, quando essa regra é desrespeitada, então temos o caso em que a **integridade referencial é violada**.

Font: <https://devmedia.com.br/integridade-de-dados-parte-02/9000>

Font: <https://pt.stackoverflow.com/questions/166120/o-que-%C3%A9-integridade-referencial>

- Descrição:
Quando estamos criando as tabelas no banco de dados podemos acabar esquecendo (principalmente no início de nosso aprendizado sobre) de colocar as chaves estrangeiras nas tabelas que necessitam das mesmas, o que, conseqüentemente, não respeita a integridade referencial, como consequência disso alguns campos não podem ser apagados sem apagar a tabela toda, isso se aplica também ao inserimento de dados ou a atualização dos mesmos, para prevenir desse tipo de situação é possível aplicar um Trigger que cria essas relações automaticamente.
 - Antes da criação das tabelas, claro!

Exemplo prático:

____PRINT

Replicação síncrona de tabelas:

O que é replicação síncrona:

- Basicamente a replicação síncrona se refere a gravação de dados em dois ou mais locais ao mesmo tempo, caso você queira preencher dois campos em duas tabelas diferentes ao mesmo tempo ou até mesmo para a criação de tabelas.

Para banco de dados simultâneos e iguais só azure aparentemente!

Exemplo prático:

PRINT

Registro de eventos e armazenamento de informações:

- Descrição: Registrar e armazenar e referem a inserir.

Exemplo prático:

PRINT

Imposição de autorizações de segurança:

- Descrição: Aplicação de uma senha para alterar, atualizar ou deletar um dado de uma tabela, basicamente um gerenciamento de permissão:

Exemplo prático:

PRINT

Impedir transações inválidas:

- Descrição: Podemos criar um Trigger que tem requisitos que devem ser preenchidos para não acionar, quando esses requisitos não são preenchidos esse Trigger aciona um “evento de impedimento” (Tipo palavras mágicas que a gente aprende quando criança, o trigger é uma pessoa fresca que está atrás da porta e você (uma transação) quer passar mas para isso precisa pedir “por favor”).

Exemplo prático:

PRINT

Fontes de Pesquisa:

Font 1:

<https://www.devmedia.com.br/triggers-no-sql-server-teoria-e-pratica-aplicada-em-uma-situacao-real/28194>

Font 2:

<https://www.alura.com.br/artigos/trigger-em-sql>