**ROTEIRO (slide)**

**O QUE SÃO TRIGGERS:**

O termo **trigger** (gatilho em inglês) define uma estrutura do banco de dados que funciona, como o nome sugere, como uma função que é disparada mediante alguma ação.

Um trigger está intimamente relacionado a uma tabela, sempre que uma dessas ações é efetuada sobre essa tabela, é possível dispará-lo para executar alguma tarefa.

É muito utilizada para ajudar a manter a consistência dos dados ou para propagar alterações em um determinado dado de uma tabela para outras. Um bom exemplo é um trigger criado para controle de quem alterou a [tabela](https://pt.wikipedia.org/wiki/Tabelas_(banco_de_dados)), nesse caso, quando a alteração for efetuada, o gatilho é "disparado" e grava em uma [tabela](https://pt.wikipedia.org/wiki/Tabelas_(banco_de_dados)) de histórico de alteração, o usuário e data/hora da alteração.

Os triggers podem ser DML ou DDL. Os DML são executados quando um usuário tentar modificar dados através de um evento de linguagem de manipulação de dados (utilização de **INSERT**, **UPDATE** ou **DELETE**). Os DDLs são executados em resposta a diversos eventos de linguagem de definição de dados (como o **CREATE**, **ALTER** ou **DROP**).

**TIPOS DE TRIGGER:**

Uma trigger pode ser de 3 tipos diferentes **FOR**, **AFTER** o **INSTEAD OF**.

Cada tipo de trigger determina o momento em que ela será executada.

* **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.

**O QUE CADA TIPO DE TRIGGER ESPECIFICA:**

* **FOR ou AFTER:** Especifica que o trigger do DML é disparado apenas quando todas as operações especificadas na instrução SQL de trigger foram iniciadas com êxito. Todas as verificações de restrição e ações referenciais em cascata também devem ter êxito para que esse gatilho seja disparado.

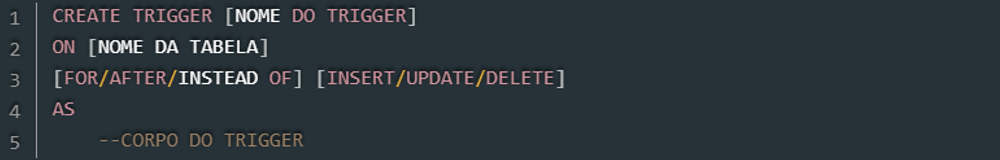
**Obs.:** Uma diferença que o AFTER tem do FOR é que **NÃO É POSSÍVEL** definir gatilhos AFTER em exibições no banco de dados (como o SELECT e o PRINT).

* **INSTEAD OF:** Especifica que o trigger DML será iniciado em vez da instrução SQL de trigger, substituindo as ações das instruções do trigger. Não é possível especificar INSTEAD OF para triggers DDL ou de logon.

**SINTAXE E ESTRUTURA BÁSICA DO TRIGGER:**

Exemplificado um DML trigger que é o mais comum, onde um dos três comandos INSERT, UPDATE ou DELETE irá disparar o trigger, ou seja, especifica as instruções de modificação de dados que, quando tentadas nessa tabela ou exibição, ativam o gatilho DML.

Pelo menos uma opção deve ser especificada e pode ser usado qualquer combinação dessas opções em qualquer ordem na definição do gatilho. Após a palavra AS temos as instruções do que o Trigger irá fazer quando disparado, na linguagem **Transact-SQL** (extensão proprietária da Microsoft e da Sybase para o SQL usado interagir com bancos de dados relacionais).

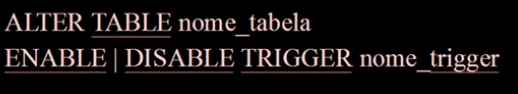


Em estruturas mais complexas de triggers podemos encontrar alguns exemplos de propriedades que podem ser adicionadas ao trigger.

* **CRIPTOGRAFIA DE DADOS** **(SEGURANÇA):** Criptografando o código do trigger (instrução WITH ENCRYPTION);
* **ANINHAMENTO:** Otrigger que ao ser disparado, pode executar uma declaração DML que leva ao disparo de outro trigger;
* **RECURSIVIDADE:** O trigger AFTER aninhado executa uma declaração DML que o dispara novamente.

**Obs.:** Os triggers recursivos podem ser habilitados ou desabilitados com o seguinte comando:



Para **HABILITAR** ou **DESABILITAR** triggers em uma tabela de uma forma geral, podemos utilizado o comando no DDL, o ALTER TABLE.

Para verificar quais triggers (por exemplo, o seu tipo ou se está habilitado ou desabilitado) estão relacionados com uma tal tabela ou com todo o banco de dados, podemos utilizar os seguintes comandos:

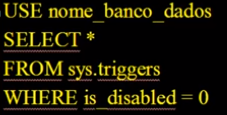


Para verificar os triggers de todo o banco de dados (instruções na cor amarela abaixo), no comando WHERE temos:

* **WHERE** **is\_disabled = 0** → Triggers que estão desabilitados;
* **WHERE** **is\_disabled = 1** → Triggers que estão habilitados.

Para descobrir quantos os triggers estão habilitados ou desabilitados em todo o banco de dados, temos o comando:

* **WHERE is\_disabled = 0 OR is\_disabled = 1**



**VANTAGENS DO USO DOS TRIGGERS:**

* Gerar alguns valores de coluna derivados automaticamente;
* Aplicar a integridade referencial;
* Registro de eventos e armazenamento de informações no acesso à tabela;
* Auditoria;
* Replicação síncrona de tabelas;
* Imposição de autorizações de segurança;
* Impedir transações inválidas.