

## O que é um Trigger SQL?

Um trigger (“gatilho”) é um objeto programável do banco de dados associado a uma tabela. Trata-se de um procedimento que é invocado automaticamente quando um comando DML é executado na tabela, sendo executado para cada linha afetada. Desta forma, as operações que podem disparar um trigger são:

- INSERT
- UPDATE
- DELETE

Geralmente, os triggers são empregados para verificar integridade dos dados, fazer validação dos dados e outras operações relacionadas.

## Diferença entre Trigger e Procedimento Armazenado

Tanto os triggers quanto as stored procedures são objetos programáveis de um banco de dados. Porém, eles possuem diferenças importantes entre si, que afetam o modo como são aplicados. Algumas das principais diferenças entre trigger e procedimentos armazenados são:

- Um Trigger é associado a uma tabela.
- Os triggers são armazenados no próprio banco de dados como arquivos separados.
- Triggers não são chamados diretamente, sendo invocados automaticamente, ao contrário dos procedimentos armazenados.
- Procedimentos armazenados podem trabalhar com parâmetros; Não passamos parâmetros aos triggers.
- Os triggers não retornam um conjunto de resultados (resultset) ao cliente chamador.

## Aplicações dos triggers

As principais aplicações dos triggers em bancos de dados são:

- Validação de Dados (tipos de dados, faixas de valores, etc).
- Rastreamento e registro de logs de atividades em tabelas.
- Verificação de integridade de dados e consistência
- Arquivamento de registros excluídos.

## Modos de Disparo de um Trigger

Um Trigger em MySQL pode ser disparado de dois modos diferentes:

- **BEFORE** – O trigger é disparado e seu código executado ANTES da execução de cada evento – por exemplo, antes de cada inserção de registros na tabela.
- **AFTER** – O código presente no trigger é executado após todas as ações terem sido completadas na tabela especificada.

## Sintaxe para criação de um trigger em MySQL

Para criar um trigger em MySQL usamos a seguinte sintaxe:

```
CREATE TRIGGER nome timing operação
ON tabela
FOR EACH ROW
declarações
```

Onde:

- **timing** pode ser BEFORE ou AFTER
- **operação** pode ser INSERT / UPDATE / DELETE

## Exemplo

Neste exemplo criaremos uma tabela chamada **Produto**, que conterá os seguintes dados:

- Nome do produto
- Identificação do produto (chave primária)
- Preço normal
- Preço com desconto a ser aplicado

Logo após criaremos um trigger de nome **tr\_desconto**, cuja função será aplicar um valor de desconto de 10% à coluna Preço\_Desconto quando for disparado. Ou seja, todos os produtos terão seu preço reduzido em 10% nesta coluna. O trigger será disparado ao inserir um novo registro na tabela, o que faremos no passo 3. Veja o código completo do exercício a seguir:

```
-- 1. Criar a tabela de exemplo:
CREATE TABLE Produto (
  idProduto INT NOT NULL AUTO_INCREMENT,
  Nome_Produto VARCHAR(45) NULL,
  Preço_Normal DECIMAL(10,2) NULL,
  Preço_Desconto DECIMAL(10,2) NULL,
  PRIMARY KEY (idProduto));

-- 2. Criar o Trigger:
CREATE TRIGGER tr_desconto BEFORE INSERT
ON Produto
FOR EACH ROW
```

```
SET NEW.Preco_Desconto = (NEW.Preco_Normal * 0.90);
```

```
-- 3. Executar uma inserção que irá disparar o Trigger:
```

```
INSERT INTO Produto (Nome_Produto, Preco_Normal)  
VALUES ("DVD", 1.00), ("Pendrive", 18.00);
```

```
-- 4. Verificar se trigger foi disparado observando o preço com  
desconto:
```

```
SELECT * FROM Produto;
```

## Como excluir um trigger

Para excluir um trigger em MySQL usamos a declaração **DROP TRIGGER**, seguida do nome do trigger, como no exemplo:

```
DROP TRIGGER tr_desconto;
```

### ATIVIDADE DE FIXAÇÃO:

PARA TABELA PRODUTOS:

CRIAR UMA COLUNA COM O NOME "DESCONTO" E CRIAR UM GATILHO QUE QUANDO VOCÊ ATUALIZA (UPDATE) NO VALOR DO PRODUTO, ELE INSIRA O MESMO VALOR COM DESCONTO A VISTA (10%) JÁ NA COLUNA DE DESCONTO.

PARA TABELA FUNCIONÁRIOS:

INSERIR NA TABELA UMA COLUNA COM O SALARIO LIQUIDO E QUANDO VOCÊ INSERIR O VALOR (INSERT) NA COLUNA SALÁRIO ELE APLICA O VALOR DO SALARIO BRUTO JÁ COM DESCONTO (DE 10%)