

ROTEIRO DE AULA PRÁTICA

TÍTULO DA AULA PRÁTICA: Trigger (*gatilho*)

Objetivos

- Entender na prática o funcionamento da *trigger* no banco de dados.
- Capacitar a resolução de problemas que requerem o uso de triggers no banco de dados.
- Realizar operações no SQL Server que disparam gatilhos para outras operações.

Trigger (*gatilho*)

Trigger ou gatilho consiste em um procedimento armazenado no banco de dados que pode ser chamado automaticamente através do disparo de um evento em espera.

De acordo com Feitosa (2013), existem diversas vantagens em transferir algumas rotinas para Triggers:

- Execução mais rápida, pois já estão pré-compilados e as queries preparadas.
- Redução do tráfego de rede, pois Triggers não recebem nenhum tipo de comando e as Procedures Armazenadas só recebem parâmetros.
- Permitem o encapsulamento da lógica do negócio.

As desvantagens são:

- A programação é específica de cada SGBD, de forma que a migração para outro SGBD obriga a reescrever todos os Triggers do banco. Isto pode ser problemático em casos em que o uso deste recurso seja amplo, pois as Triggers são programas e portanto sujeitos a bugs.
- Em alguns SGBDs, os Triggers podem ser desativados pelo Administrador do Banco e aquela determinada funcionalidade para de funcionar.

O Trigger é disparada mediante alguma ação, geralmente, alterações nas tabelas por meio de operações de inserção, exclusão e atualização de dados (insert, delete e update). Desta forma, o Trigger está intimamente relacionado a uma tabela.

No SQL Server, utilizamos instruções DML (*Data Manipulation Language*) para criar, alterar ou excluir um trigger.

A sintaxe para criação de um trigger é a seguinte:

```

CREATE TRIGGER [NOME DO TRIGGER]
ON [NOME DA TABELA]
[FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]
AS
    --CORPO DO TRIGGER

```

Os parâmetros são:

- **NOME DO TRIGGER:** nome que identificará o gatilho como objeto do banco de dados. Deve seguir as regras básicas de nomenclatura de objetos.
- **NOME DA TABELA:** tabela à qual o gatilho estará ligado, para ser disparado mediante ações de insert, update ou delete.
- **FOR/AFTER/INSTEAD OF:** uma dessas opções deve ser escolhida para definir o momento em que o trigger será disparado. FOR é o valor padrão e faz com o que o gatilho seja disparado junto da ação. AFTER faz com que o disparo se dê somente após a ação que o gerou ser concluída. INSTEAD OF faz com que o trigger seja executado no lugar da ação que o gerou.
- **INSERT/UPDATE/DELETE:** uma ou várias dessas opções (separadas por vírgula) devem ser indicadas para informar ao banco qual é a ação que disparará o gatilho. Por exemplo, se o trigger deve ser disparado após toda inserção, deve-se utilizar AFTER INSERT.

Considere o Diagrama de Entidade e Relacionamento do ecommerce, é possível imaginarmos o gatilho de atualização do valor do pedido quando é inserido um novo item de pedido.

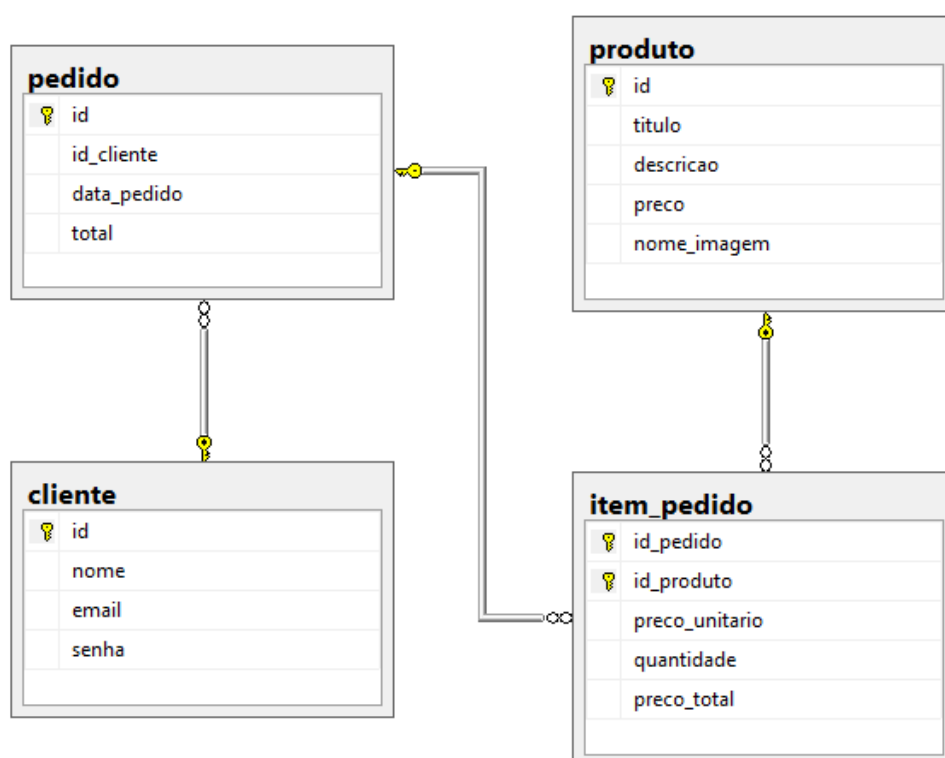


Figura 1: Diagrama de entidade e relacionamento de um sistema para *ecommerce*.

Por exemplo, o trigger irá disparar quando for inserido um novo registro no item de pedido e atualizará o valor total do pedido.

```
create trigger atualizaTotalPedido
on [item_pedido]
after insert
as
begin
    declare
        @valor decimal(6,2),
        @id_pedido int

    select @id_pedido = id_pedido, @valor = preco_total from inserted

    update pedido set total = total + @valor
    where id = @id_pedido
end
```

Para testar, busque um determinado pedido e seus itens de pedidos na base de dados.

```
select * from pedido where id = 1
select * from item_pedido where id_pedido = 1
```

Observe que existem dois itens de pedidos e o valor total do pedido corresponde à soma dos preços totais dos itens de pedidos.

Resultados		Mensagens			
	id	id_cliente	data_pedido	total	
1	1	1	2017-04-11 00:00:00.000	3799.00	

	id_pedido	id_produto	preco_unitario	quantidade	preco_total
1	1	1	2799.00	1	2799.00
2	1	2	1000.00	1	1000.00

Insira um novo item de pedido para o mesmo pedido.

```
insert into item_pedido values (1, 3, 1000, 1, 1000)
```

Observe que duas linhas são afetadas ao executar a inclusão do novo item de pedido. Isto ocorre, pois a trigger será disparada.

O trigger atualiza automaticamente o valor total do pedido de acordo com os itens de pedidos.

Resultados		Mensagens	
id	id_cliente	data_pedido	total
1	1	2017-04-11 00:00:00.000	4799.00

	id_pedido	id_produto	preco_unitario	quantidade	preco_total
1	1	1	2799.00	1	2799.00
2	1	2	1000.00	1	1000.00
3	1	3	1000.00	1	1000.00

Exercícios

1. Utilizando o DER da Figura 1:
 - a) Crie um Trigger para atualizar o valor total do pedido quando um item de pedido for atualizado (*update*).
 - b) Sempre que modificar a quantidade de itens de pedido, atualize o valor realizando uma “regra de três” com os valores anteriores.

Referência Bibliográfica

FEITOSA, M. P. **Fundamentos de Banco de Dados**. Uma abordagem prático-didática. ed. 1, São Paulo: Independente, 2013.

Triggers no SQL Server: teoria e prática aplicada em uma situação real.

Disponível em: < <https://www.devmedia.com.br/triggers-no-sql-server-teoria-e-pratica-aplicada-em-uma-situacao-real/28194>>. Acessado em: 14 nov. 2023.