

Curso de Especialização:
Engenharia e Administração de Sistemas de Banco de Dados

Fundamentos de Sistemas de Banco de Dados



Transact SQL – Gatilhos

Profa. Dra. Gisele Busichia Baioco

gisele@ft.unicamp.br



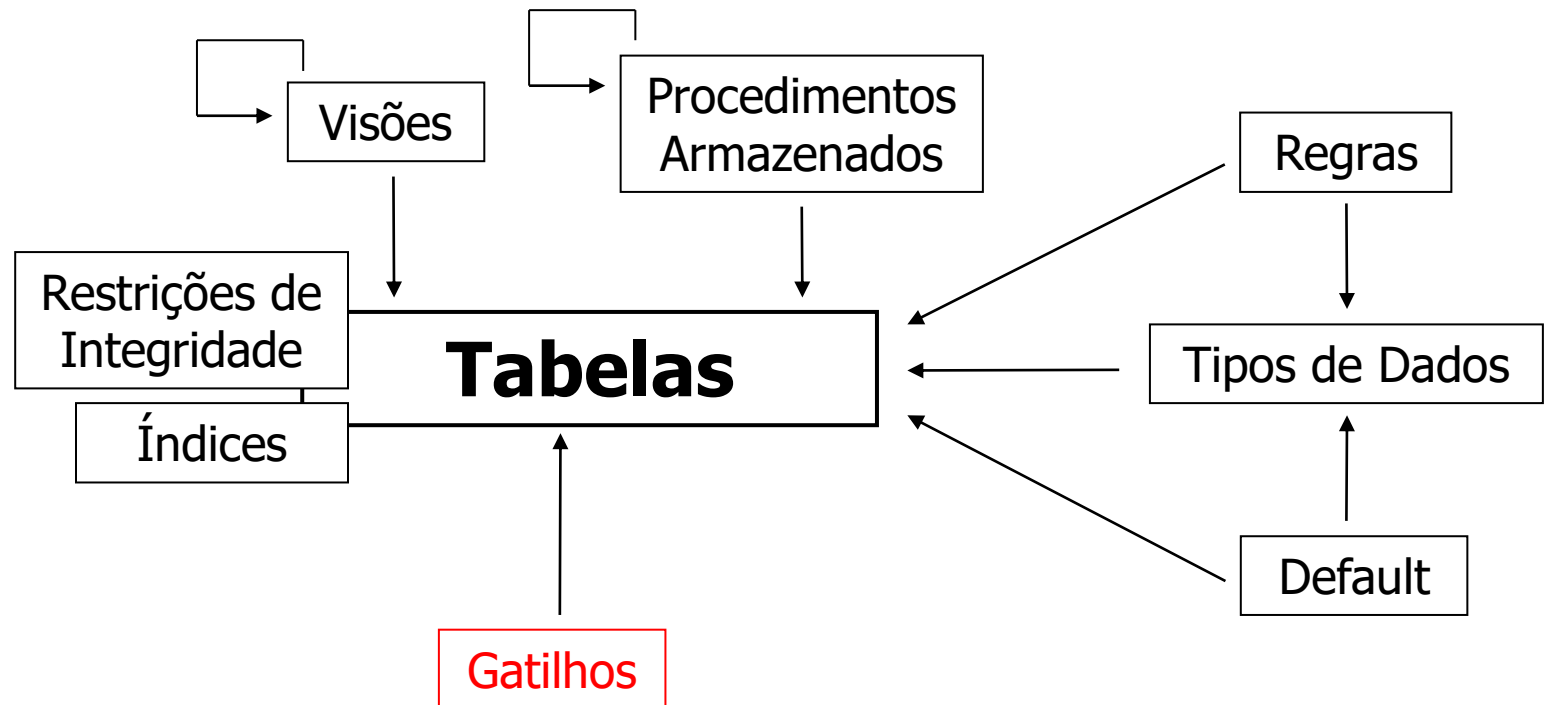
UNICAMP



FACULDADE DE TECNOLOGIA

Gatilhos – *Triggers*

Banco de Dados





Gatilhos – *Triggers*

- **Definição:** um gatilho (*trigger*) é um conjunto de comandos SQL associado a uma tabela, que é executado automaticamente pelo SGBD quando os dados da tabela forem modificados (*insert*, *update* ou *delete*).
 - Um gatilho pode ser visto como um tipo de procedimento armazenado que não possui parâmetros e que é executado automaticamente pelo SGBD quando ocorre algum tipo de alteração na tabela a qual esteja vinculado.
- **Observações:**
 - Um gatilho não pode ser chamado explicitamente;
 - Sempre que o comando SQL (*insert*, *update* ou *delete*) associado a um gatilho for executado, o gatilho é **automaticamente** executado;
 - A execução de um gatilho é repetida para cada linha (registro) afetada pelo comando SQL associado;
 - Os gatilhos são designados para fazer verificações ou modificar dados. Assim, não devem retornar dados ao usuário (*select*).



Gatilhos – *Triggers*

- **Utilização convencional:**

- manutenção de integridade (quando não contemplada por outros recursos do SGBD: integridade de chave – UNIQUE; integridade de entidade – PRIMARY KEY e NOT NULL; integridade referencial – FOREIGN KEY);
- modificação de dados em cascata entre tabelas relacionadas entre si.

- **Outra utilização – criação de BDs Ativas:**

- Um sistema de BDs Ativas não apenas armazena dados e executa transações de usuários, mas também executa ações em resposta a eventos, como alterações dos dados.
- As **Regras Ativas** (implementadas pelos gatilhos) especificam quando e quais ações deverão ser executadas.
- O modelo *evento-condição-ação* é utilizado para a implementação de regras ativas:

```
on evento  
if condição  
then ação
```



Gatilhos – *Triggers*

■ Funcionamento:

- Cada comando SQL é considerado uma transação pelo MS-SQL Server. Caso uma tabela possua um gatilho para um dado comando SQL, a transação é estendida com os comandos codificados no gatilho. Por exemplo, o seguinte comando SQL:

```
delete from tabela  
where coluna = valor
```

É considerado pelo SQL Server como:

```
begin transaction  
    escreve linha(s) no log  
    remove linha(s) da tabela  
    executa gatilho de remoção  
commit/rollback transaction
```

- Como o código de um gatilho pertence à transação definida pelo comando SQL, as alterações efetuadas pelo gatilho estão sujeitas a um cancelamento por meio de um comando *rollback* (em um *batch* ou procedimento armazenado, por exemplo).



Gatilhos – *Triggers*

Como criar um gatilho usando T-SQL?

- Antes de criar deve-se projetar um mecanismo de gatilho:
 1. Especificando as condições sobre as quais o gatilho deve ser executado (quais operações de modificação em quais tabelas);
 2. Especificando as ações que serão tomadas quando o gatilho for disparado.
- Comando CREATE TRIGGER;
- Sintaxe básica:

```
create trigger nome_gatilho  
on nome_tabela  
for {[insert] [,] [update] [,] [delete]}  
as  
comandos_SQL
```



Gatilhos – *Triggers*

Como criar um gatilho usando T-SQL?

- Exemplo: Considere o seguinte esquema físico da BD.

```
create table produto (  
  codproduto int not null,  
  nome varchar(60) not null,  
  preco numeric(5,2) not null,  
  primary key (codproduto)  
)  
go  
create table notafiscal (  
  numeronota numeric(10,0) not null,  
  valortotal numeric(10,2) not null,  
  primary key (numeronota)  
)  
go
```

```
create table itemnotafiscal (  
  numeronota numeric(10,0) not null,  
  codproduto int not null,  
  quantidade int not null,  
  primary key (numeronota,codproduto),  
  foreign key (numeronota) references notafiscal,  
  foreign key (codproduto) references produto  
)  
go
```

- Gatilhos a serem criados:
 1. Quando for inserido um item, a coluna **valortotal** será atualizada, na tabela **notafiscal**;
 2. Quando for excluída uma nota fiscal, todos os seus itens serão excluídos automaticamente;
 3. Quando for alterada a quantidade de um item, o valor total da nota fiscal deve ser alterado também.



Gatilhos – *Triggers*

Exemplo: gatilho para inserção

■ Quando é feita a inserção de uma ou mais linhas em uma tabela, o SQL Server cria uma tabela lógica (em memória) chamada **inserted** com as seguintes características:

- Contém as linhas que serão inseridas (mas ainda não foram);
- Tem a mesma estrutura da tabela real;
- Pode-se consultar dados nessa tabela com o comando *select*, da mesma maneira que em uma tabela real.

```
create trigger inclusaoitemnota
on itemnotafiscal for insert
as
    update notafiscal
    set valortotal = valortotal + (select i.quantidade * p.preco
                                   from produto p inner join inserted i
                                   on p.codproduto = i.codproduto)
    where numeronota = (select numeronota from inserted)
    if @@rowcount = 0
        rollback transaction
```




Gatilhos – *Triggers*

Exemplo: gatilho para exclusão

■ Quando é feita a exclusão de uma ou mais linhas em uma tabela, o SQL Server cria uma tabela lógica (em memória) chamada **deleted** com as seguintes características:

- Contém as linhas que serão excluídas (mas ainda não foram);
- Tem a mesma estrutura da tabela real;
- Pode-se consultar dados nessa tabela com o comando *select*, da mesma maneira que em uma tabela real.

```
create trigger exclusivaonota
on notafiscal for delete
as
    -- excluir todos os itens relacionados (mesmo numeronota que deleted)
    delete from itemnotafiscal
    where numeronota in (select numeronota from deleted)
    if @@rowcount = 0
        rollback transaction
```



Gatilhos – *Triggers*

Exemplo: gatilho para atualização

- Pode-se considerar uma atualização como uma exclusão seguida de uma inserção (excluem-se valores antigos e inserem-se valores novos):
 - A tabela **inserted** contém os dados atualizados que serão inseridos na tabela real;
 - A tabela **deleted** os dados antigos, que estão sendo excluídos.

```
create trigger alteracaoitemnota
on itemnotafiscal for update
as
if update(quantidade)
begin
    update notafiscal
    set valortotal = valortotal + (select p.preco * (i.quantidade - d.quantidade)
                                   from produto p inner join inserted i
                                   on p.codproduto = i.codproduto
                                   inner join deleted d
                                   on i.codproduto = d.codproduto and
                                   i.numeronota = d.numeronota)

    where numeronota = (select numeronota from inserted)
    if @@rowcount = 0
        rollback transaction
end
```



Gatilhos – *Triggers*

Testes dos gatilhos

- Inserção de alguns produtos:

```
insert into produto values (1, 'lapis', 1.50)
```

```
insert into produto values (2, 'borracha', 2.00)
```

```
insert into produto values (3, 'caneta', 3.50)
```

- Inserção de uma nota fiscal:

```
insert into notafiscal values (1, 0)
```

- Inserção de itens de nota fiscal (que executam o gatilho de inserção):

```
insert into itemnotafiscal values (1, 2, 10)
```

```
insert into itemnotafiscal values (1, 3, 2)
```



Gatilhos – *Triggers*

Testes dos gatilhos

- Os próximos comandos resultarão em erro de restrição de integridade (*foreign key*):

```
insert into itemnotafiscal values (0, 1, 5)
```

```
delete from notafiscal where numeronota = 1
```

- Para que o gatilho de exclusão seja executado com sucesso é necessário alterar a tabela **itemnotafiscal**, excluindo a restrição de chave estrangeira referente à tabela **notafiscal**:

```
alter table itemnotafiscal
```

```
drop constraint <nome_restricção>
```

onde <nome_restricção> pode ser obtido com o procedimento do sistema **sp_help**:

```
sp_help itemnotafiscal
```



Gatilhos – *Triggers*

Testes dos gatilhos

- Devido a exclusão da restrição de chave estrangeira referente à tabela **notafiscal**, é necessário alterar também o gatilho de inserção, colocando um trecho de código no início que testa se existe uma nota para a inclusão de um item de nota:

```
alter trigger inclusaoitemnota
on itemnotafiscal for insert
as
    if not exists (select *
                  from inserted i inner join notafiscal n
                  on i.numeronota = n.numeronota)
begin
    print 'Esse item não contém um número de nota válido'
    rollback transaction
end
else begin
    update notafiscal
    set valortotal = valortotal + (select i.quantidade * p.preco
                                   from produto p inner join inserted i
                                   on p.codproduto = i.codproduto)
    where numeronota = (select numeronota from inserted)
    if @@rowcount = 0
        rollback transaction
end
```



Gatilhos – *Triggers*

Testes dos gatilhos

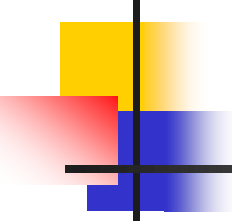
- Para verificar os efeitos deve-se executar novamente:

```
insert into itemnotafiscal values (0, 1, 5)
delete from notafiscal where numeronota = 1
```

- Para testar o gatilho de atualização, basta incluir novamente produtos, nota fiscal e itens de notafiscal e, então, executar:

```
update itemnotafiscal set quantidade = 5
where numeronota = 1 and codproduto = 2
```

```
update itemnotafiscal set quantidade = 4
where numeronota = 1 and codproduto = 3
```



Gatilhos – *Triggers*

Como alterar um gatilho?

- Comando ALTER TRIGGER;
- Sintaxe básica:

```
alter trigger nome_gatilho
on nome_tabela
for {[insert] [,] [update] [,] [delete]}
as
comandos_SQL
```

- Esse comando sobrepõe a definição do gatilho original;
- Para alterar apenas o nome de um gatilho, deve-se utilizar o procedimento armazenado do sistema **sp_rename**.
 - Sintaxe:

```
sp_rename nome_velho, nome_novo
```



Gatilhos – *Triggers*

Como excluir um gatilho?

- Comando DROP TRIGGER;

- Sintaxe:

```
drop trigger nome_gatilho1 [,... nome_gatilhon]
```

- Exemplo:

```
drop trigger inclusaoitemnota, alteracaoitemnota
```




Bibliografia

ELMASRI, R.; NAVATHE, S. B., Fundamentals of database systems. 7 ed., Pearson, 2016.

Triggers:

<https://docs.microsoft.com/pt-br/sql/relational-databases/triggers/dml-triggers?view=sql-server-ver15>

<https://docs.microsoft.com/pt-br/sql/relational-databases/triggers/create-dml-triggers-to-handle-multiple-rows-of-data?view=sql-server-ver15>

<https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver15>

Linguagem de controle de fluxo de execução:

<https://docs.microsoft.com/pt-br/sql/t-sql/language-elements/control-of-flow?view=sql-server-ver15>