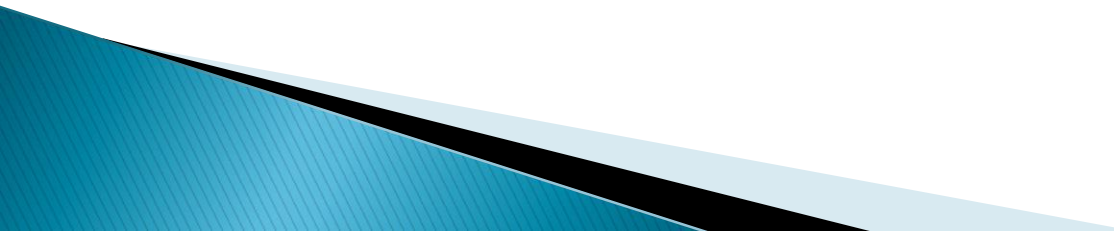



TRIGGERS (Gatilhos)

Prof. Anésio Freire

Triggers – Definição

- ▶ Utilizados para implementar regras de negócio da aplicação;
 - ▶ Procedimento disparado pelo SGBD em resposta a um evento específico do banco de dados;
 - ▶ O evento é associado à tentativa de execução de operações sobre uma tabela (inclusão, exclusão e alteração);
- 

Triggers – Para que serve?

- ▶ Preenchimento de campo resultante de uma expressão;
 - ▶ Crítica aos dados com mensagens mais adequadas às regras do negócio;
 - ▶ assegurar as restrições de integridade
 - ▶ melhorar a segurança no acesso aos dados
 - ▶ melhorar o desempenho das aplicações
 - ▶ reduzir o volume de dados trafegados na rede
 - ▶ melhorar a manutenção do software.
- 

Triggers – Funcionamento

- ▶ Quando um trigger é executado, o SQL Server cria **duas tabelas temporárias** que ficam residentes na memória e existem apenas enquanto o trigger estiver ativo.
- ▶ As duas tabelas contêm a mesma estrutura da tabela de base onde o trigger foi criado.
 - Essas duas tabelas recebem o nome de **INSERTED** e **DELETED**.
 - Você pode consultar dados nessa tabela com o SELECT, da mesma forma que uma tabela real

Triggers – Evento

- ▶ O que faz o gatilho disparar?
- ▶ O gatilho é disparado por um dos seguintes eventos do banco de dados, dependendo de como foi definido:
 - INSERT
 - UPDATE
 - DELETE

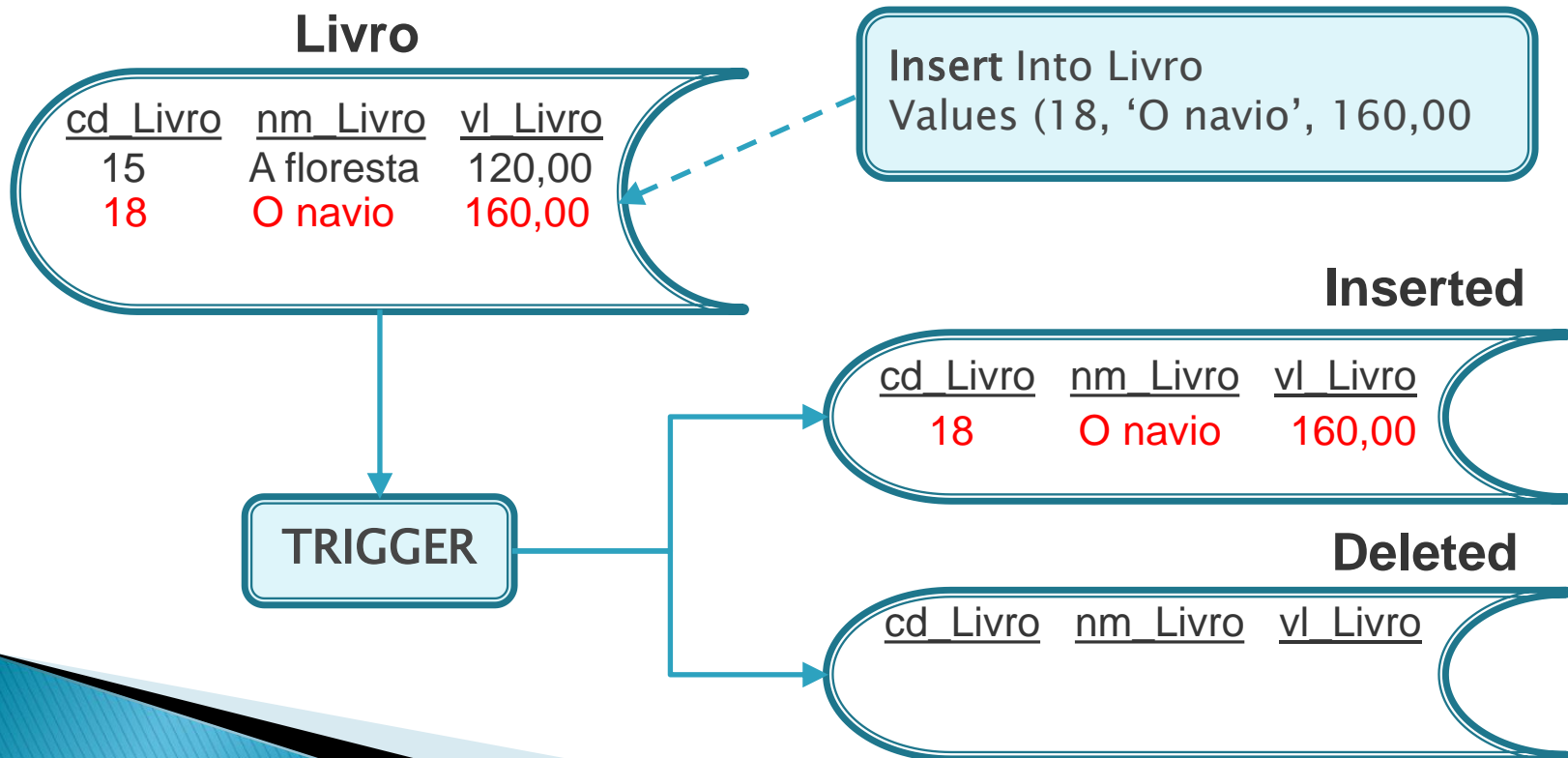
Triggers After – Sintaxe

```
CREATE TRIGGER <nome> ON <tabela>  
FOR [INSERT][,][UPDATE][,][DELETE]  
AS <comandos SQL>
```

After: Um Trigger AFTER é executado depois que ação das instruções INSERT, UPDATE ou DELETE for executada. Especificar AFTER é o mesmo que especificar FOR, que é a única opção disponível em versões anteriores do Microsoft SQL Server.

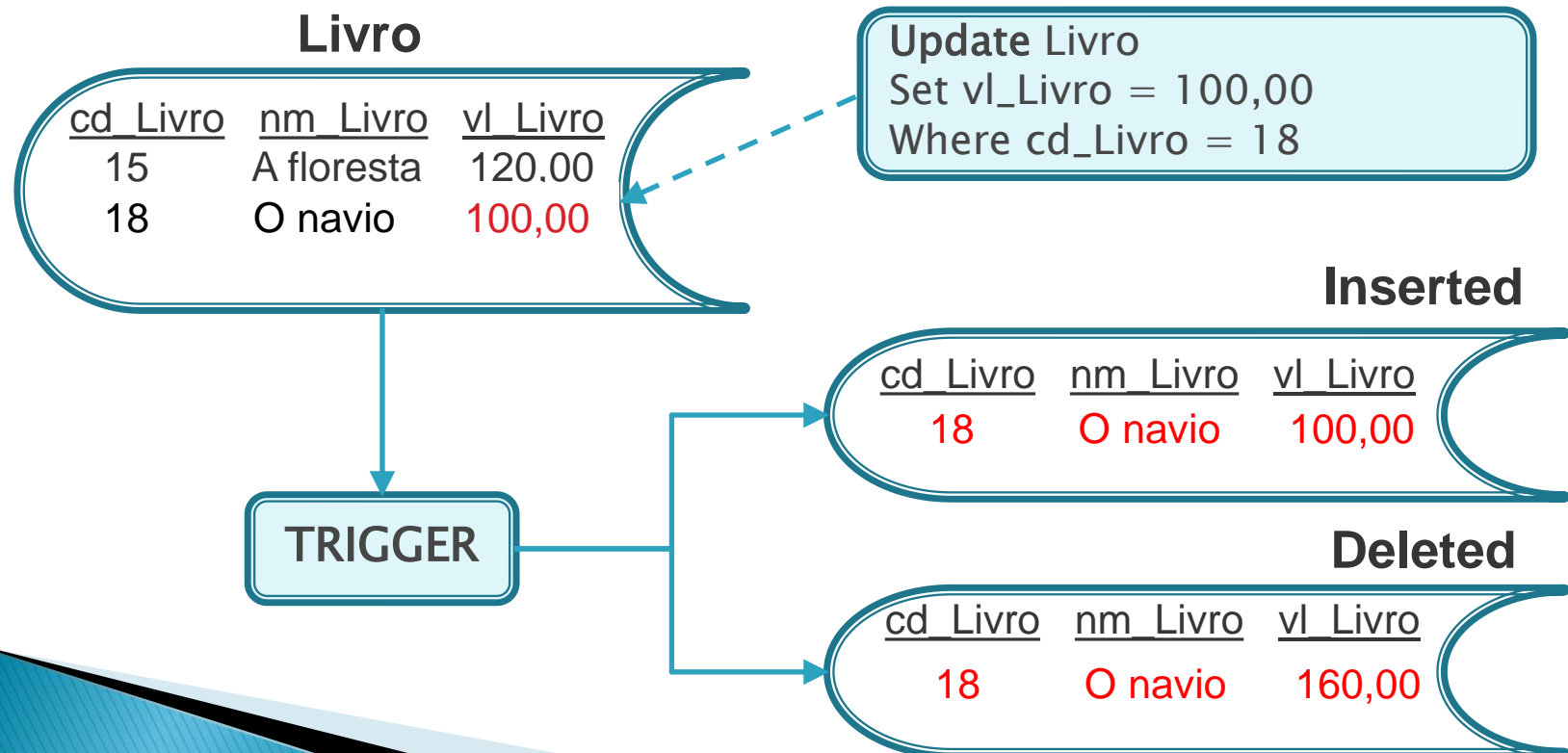
Triggers – Funcionamento

- Quando um comando **INSERT** é executado, o registro criado será copiado para a tabela de base e para a tabela INSERTED.



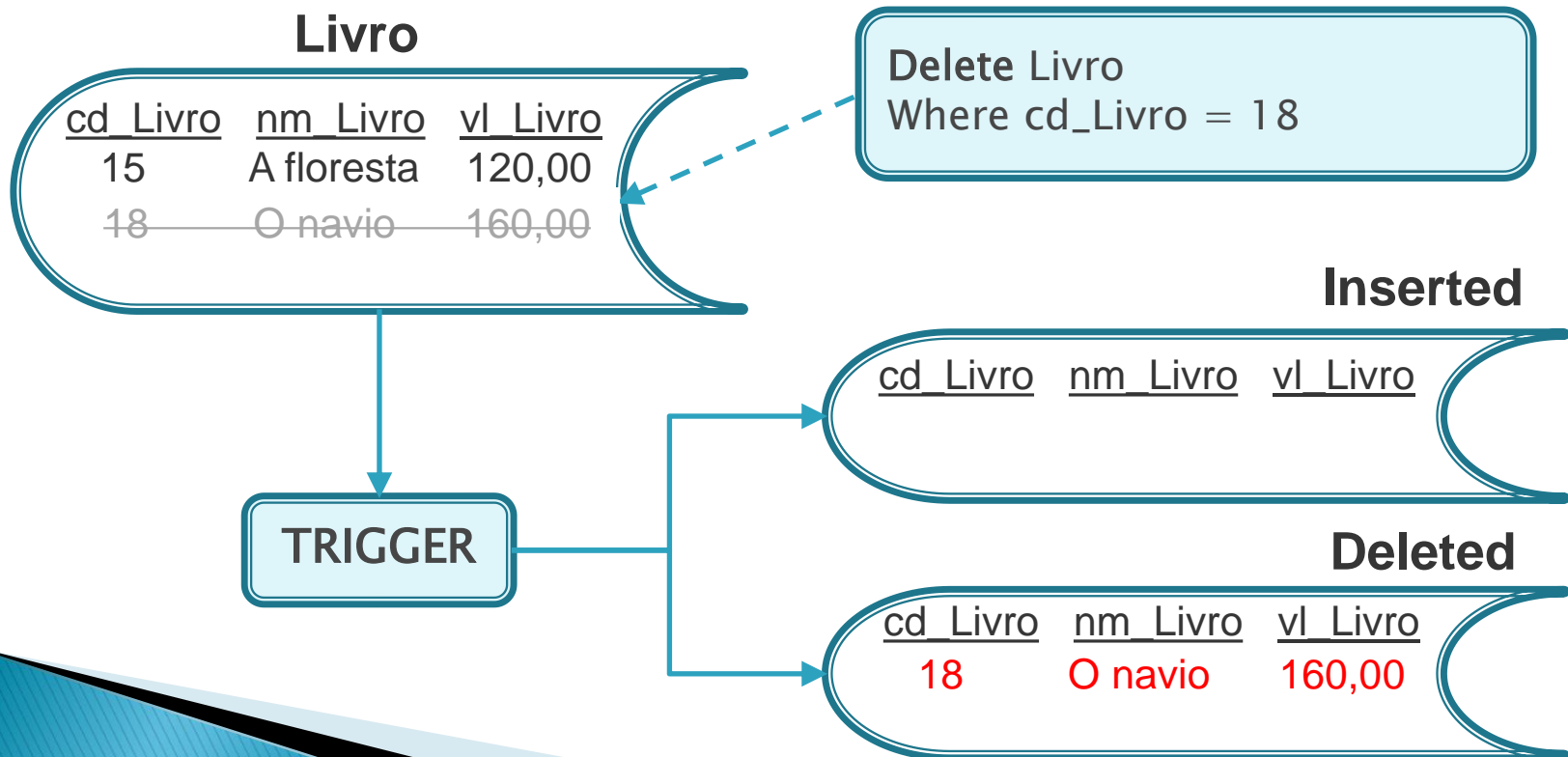
Triggers – Funcionamento

- Quando um comando **UPDATE** é executado, o registro original (registro antigo) é armazenado na tabela DELETED e o registro modificado (registro atual) é armazenado na tabela de base e na tabela INSERTED.

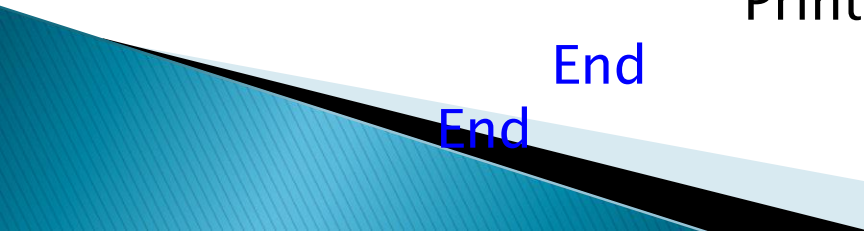


Triggers – Funcionamento

- ▶ Quando um comando **DELETE** é executado, o registro excluído é armazenado na tabela DELETED.



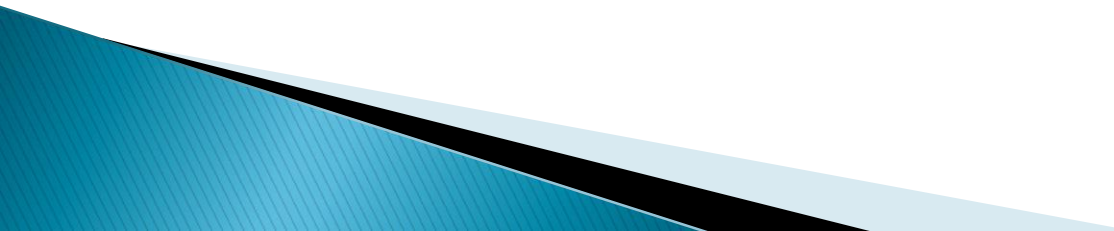
```
Create Trigger Exemplo On ExemploTrigger
For Insert, Update, Delete
As
Begin
    If Exists (Select Id From Deleted)
    Begin
        If Exists (Select Id From Inserted)
        Begin
            Print 'Foi feito um Update'
        End
        Else
        Begin
            Print 'Foi feito um Delete'
        End
    End
    Else
    Begin
        Print 'Foi feito um insert'
    End
End
```



Triggers Instead Of – Sintaxe

```
CREATE TRIGGER <nome> ON <tabela>  
INSTEAD OF [INSERT][,][UPDATE][,][DELETE]  
AS <comandos SQL>
```

INSTEAD OF: Pode ser executado no lugar da ação de um gatilho usual (after). Outra propriedade interessante é que eles **ignoram incondicionalmente** o evento (INSERT, UPDATE ou DELETE) executado.



Triggers – Exemplo

Criar as Tabelas

Estoque (cd_Prod, nm_Prod, qt_Prod, qt_Minima)

Compras (cd_Comp, *cd_Prod*, qt_Prod)

Vendas (cd_Venda, *cd_Prod*, qt_Prod)

Triggers – Exemplo

- ▶ Quando um produto é comprado, acrescenta-se um item do produto no estoque (O estoque é atualizado automaticamente)

```
Create Trigger tr_Compras On Compras
For Insert As
Begin
    Update Estoque
    Set E.qt_Prod = E.qt_Prod + I.qt_Prod
    From Estoque E, Inserted I
    Where I.cd_Prod = E.cd_Prod
End
```

Triggers – Exemplo

- ▶ Ao vender um produto, a quantidade em estoque deve ser decrementada. E deve-se comparar com sua quantidade mínima em estoque.

```
Create Trigger tr_Vendas On Vendas
For Insert As
Begin
    Update Estoque
    Set E.qt_Prod = E.qt_Prod - I.qt_prod
    From Estoque E, Inserted I Where I.cd_prod= E.cd_Prod
    If Exists (Select cd_Prod From Estoque Where qt_Prod <=
qt_Minima)
        PRINT 'Existem Produtos em Falta'
End
```

Triggers – Requisitos

- ▶ Especificar quando o trigger deve ser executado;
 - Isso é desmembrado em um evento que faz com que o trigger seja verificado e uma condição que precisa ser satisfeita para que a execução do trigger prossiga.
- ▶ Regras ECA
 - Evento
 - Condição – a ser satisfeita na presença do evento E
 - Ação – a ser tomada caso a condição C seja satisfeita

Triggers – Exemplos

- ▶ Supondo que um cliente de um banco faça uma retirada de dinheiro maior do que o seu saldo, o BD deveria agir da seguinte forma:
 - Seta para zero o saldo do cliente;
 - Cria um empréstimo no valor da dívida para o cliente;
 - **Evento**: operação de modificação do saldo
 - **Condição**: se $\text{saldo} < 0$
 - **Ação**: Criar um empréstimo para conta
conta (cd_Conta, nm_Titular, vl_Saldo)
emprestimo (cd_Emp, cd_Conta, vl_Emp)

Create Trigger tr_EmprestimoAuto on conta

For Update as

begin

 If exists (select cd_Conta from inserted where vl_Saldo < 0)

 begin

 Declare @numC int, @numE int, @valor money

 Set @numC = (select cd_Conta from inserted
 where vl_saldo <= 0)

 Set @numE = (select max(cd_Emp) from Emprestimo) + 1

 Set @valor = (select abs(vl_Saldo) from inserted)

 Insert into emprestimo values (@numE, @numC, @valor)

 Update conta set vl_Saldo = 0 where cd_Conta = @numC

 Print 'Foi feito um empréstimo automático'

 end

end



Observações Importantes

- ▶ Triggers não podem ser criadas para **Visões** ou **tabelas temporárias**
- ▶ Triggers **não possuem parâmetros** e não podem ser explicitamente invocados.
 - São disparados quando os dados da “**tabela protegida**” são modificados
- ▶ Triggers são considerados como parte de uma transação
 - Se houver falha no seu funcionamento, os comandos serão revertidos (ROLLBACK)