



Banco de Dados

Triggers no SQL Server

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

7. Restrições de Integridade

- Triggers -

❑ Sintaxe da cláusula CREATE TRIGGER (SQL Server)

```
CREATE TRIGGER nome_trigger ON nome_tabela
```

```
{ [FOR / AFTER / INSTEAD OF]
```

```
    { [DELETE] [,] [INSERT] [,] [UPDATE] } [NOT FOR  
    REPLICATION]
```

```
    AS
```

```
        sentença [ {sentença ... } ] }
```

```
| [FOR / AFTER / INSTEAD OF]
```

```
    { [INSERT] [,] [UPDATE] } [NOT FOR REPLICATION]
```

```
    AS
```

```
        { IF UPDATE (nome_coluna)
```

```
            [{AND | OR} UPDATE (nome_coluna)]
```

```
            [ { {AND | OR} UPDATE (nome_coluna) ... } ]
```

```
            sentença [ {sentença ... } ] }
```

```
}
```

7. Restrições de Integridade

- Triggers -

- ❑ Sintaxe da cláusula CREATE TRIGGER (SQL Server)
 - ↳ NOT FOR REPLICATION
 - ⇒ Trigger não deve ser disparado quando o mecanismo de replicação altera a tabela envolvida no trigger
 - ↳ Sentença
 - ⇒ Especifica a condição e ação do trigger
 - ⇒ Sentença SQL
 - Update
 - ⇒ Sentença da linguagem proprietária do SGBD
 - T-SQL (SQL Server)
 - PL/SQL (Oracle)
 - ↳ UPDATE(nome_coluna)
 - ⇒ Indica que a ação do trigger deve ser disparada para uma atualização sobre a coluna nome_coluna

7. Restrições de Integridade

- Triggers -

❑ Exemplos

↳ Garantir que menor salário de empregado será 1000
create trigger tr1 on Empregado for insert,update
as

```
if (select salario from inserted)< 1000
begin
    raiserror('salario invalido', 16,1)
    rollback transaction
end
```

↳ Observação importante

⇒ O SQL Server implementa as tabelas temporárias

⇒ **Inserted**

➤ Armazena cópias das tuplas afetadas pelos comandos
+ Insert e update

⇒ **Deleted**

➤ Armazena cópias das tuplas afetadas pelo comando
+ delete

7. Restrições de Integridade

- Triggers -

```
CREATE TRIGGER Sal_Max_Depart ON `FUNCIONARIO`  
FOR INSERT, UPDATE  
AS  
BEGIN  
    DECLARE @NEW_SALARIO DECIMAL(7,2);  
    SELECT @NEW_SALARIO = Salario FROM Inserted;  
    IF (@NEW_SALARIO)+(SELECT Sal_total FROM DEPARTAMENTO  
NATURAL JOIN Inserted) > 15000  
    BEGIN  
        RAISERROR('Erro: Salário máximo atingido', 16, 1);  
        ROLLBACK TRANSACTION;  
    END  
    ELSE  
        UPDATE DEPARTAMENTO SET Sal_total = Sal_total +  
@NEW_SALARIO  
END
```

7. Restrições de Integridade

- *Triggers* -

```
CREATE TRIGGER Sal_Ger_Depart ON `FUNCIONARIO`  
FOR INSERT, UPDATE  
AS  
BEGIN  
    DECLARE @NEW_SALARIO DECIMAL(7,2);  
    SELECT @NEW_SALARIO = Salario FROM Inserted;  
    IF (@NEW_SALARIO > SELECT G.Salario FROM FUNCIONARIO G  
    INNER JOIN (DEPARTAMENTO D NATURAL JOIN Inserted) ON G.cpf =  
    D.Cpf_gerente)  
        BEGIN  
            RAISERROR('Erro: Salário máximo atingido', 16, 1);  
            ROLLBACK TRANSACTION;  
        END  
    ELSE  
        UPDATE DEPARTAMENTO SET Sal_total = Sal_total +  
        @NEW_SALARIO  
END
```

7. Restrições de Integridade

- Triggers -

■ Cláusula CREATE TRIGGER

- Observações: (1/3)
 - A cláusula *referencing old row as* pode ser usada para criar uma variável armazenando o valor antigo de uma linha atualizada ou excluída. A cláusula *referencing new row as* pode ser usada com inserções além de atualizações;
 - Triggers podem ser ativadas antes (*before*) do evento (*insert/delete/update*), ao invés de depois (*after*) do evento:
 - Servem como restrições extras para impedir atualizações inválidas.

7. Restrições de Integridade

- Triggers -

❑ Cláusula CREATE TRIGGER

- Observações: (2/3)
 - Em vez de executar uma ação para cada linha afetada, podemos executar uma única ação para a instrução SQL inteira que causou o *insert/delete/update*. Para fazer isso, usamos a cláusula *for each statement* no lugar da cláusula *for each row*.
 - As cláusulas *referencing old table as* ou *referencing new table as* podem ser usadas para referenciar tabelas temporárias (tabelas de transição). As tabelas de transição não podem ser usadas com triggers *before*, somente com triggers *after*.

7. Restrições de Integridade

- Triggers -

❑ Cláusula CREATE TRIGGER

- Observações: (3/3)
 - Triggers podem ser desativados ou ativados; como padrão, eles são ativados quando criados, mas podem ser desativados por meio de *alter trigger nome_trigger disable* (ou *disable trigger nome_trigger*, dependendo do banco).
 - Um trigger pode ser removido permanentemente usando o comando *drop trigger nome_trigger*.
 - Para atualizações, o trigger pode especificar colunas cuja atualização faz com que o mesmo seja executado:
 - *create trigger tg_sld_dev after update of saldo on conta*



Banco de Dados

Excessões nos Triggers no MySQL

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

7. Restrições de Integridade

- *SIGNAL* -

❑ Levantando uma Exceção [Forma Geral]:

```
SIGNAL condition_value  
[SET signal_information_item  
[, signal_information_item] ...]
```

```
condition_value: {  
    SQLSTATE [VALUE] sqlstate_value  
    | condition_name  
}
```

❑ Geralmente:

```
SIGNAL SQLSTATE [VALUE] sqlstate_value  
[SET signal_information_item = value_1 [, signal_information_item =  
value_2, etc;]]
```

7. Restrições de Integridade

- *SIGNAL* -

❑ Levantando uma Exceção [Forma Geral]:

signal_information_item:

condition_information_item_name = simple_value_specification

condition_information_item_name: {

CLASS_ORIGIN

| SUBCLASS_ORIGIN

| CONSTRAINT_CATALOG

| CONSTRAINT_SCHEMA

| CONSTRAINT_NAME

| CATALOG_NAME

| SCHEMA_NAME

| TABLE_NAME

| COLUMN_NAME

| CURSOR_NAME

| MESSAGE_TEXT

| MYSQL_ERRNO

}

7. Restrições de Integridade

- SIGNAL -

❑ Signal Condition Information Items [Definição]:

Item Name	Definition
CLASS_ORIGIN	VARCHAR(64)
SUBCLASS_ORIGIN	VARCHAR(64)
CONSTRAINT_CATALOG	VARCHAR(64)
CONSTRAINT_SCHEMA	VARCHAR(64)
CONSTRAINT_NAME	VARCHAR(64)
CATALOG_NAME	VARCHAR(64)
SCHEMA_NAME	VARCHAR(64)
TABLE_NAME	VARCHAR(64)
COLUMN_NAME	VARCHAR(64)
CURSOR_NAME	VARCHAR(64)
MESSAGE_TEXT	VARCHAR(128)
MYSQL_ERRNO	SMALLINT UNSIGNED

PS: O padrão de caracteres (character set) para itens tipo character é UTF-8.

7. Restrições de Integridade

- SIGNAL -

❑ Signal Condition Information Items [Definição]:

Não é permitido associar o valor NULL para um '*condition information item*' em uma sentença SIGNAL.

Uma sentença SIGNAL sempre especifica um valor SQLSTATE. Os dois primeiros caracteres de um valor SQLSTATE são a sua classe (Class):

- Class = '00' (success)
Ilegal, pois este valor indica sucesso e não é válido para SIGNAL.
- Class = '01' (warning)
O valor da variável de sistema *warning_count* é incrementado.
- Class = '02' (not found)
NOT FOUND handlers catch the signal.
- Class > '02' (exception)
SQLEXCEPTION handlers catch the signal.
- Class = '40'
Tratado como uma exceção comum.

7. Restrições de Integridade

- Exemplo 1) *Trigger* com SIGNAL no MySQL -

❑ Levantando uma Exceção [Exemplo]:

```
CREATE TRIGGER before_insert_Estudante_Idade BEFORE INSERT ON
Estudante_Idade
FOR EACH ROW
BEGIN
    IF NEW.idade < 0
    THEN SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Idade menor do que 0';
    END IF;
END; //
```

```
mysql> INSERT INTO Estudante_Idade(idade, nome) VALUES (-
10,'Paulo')//
ERROR 1644 (45000): Idade menor do que 0
```

7. Restrições de Integridade

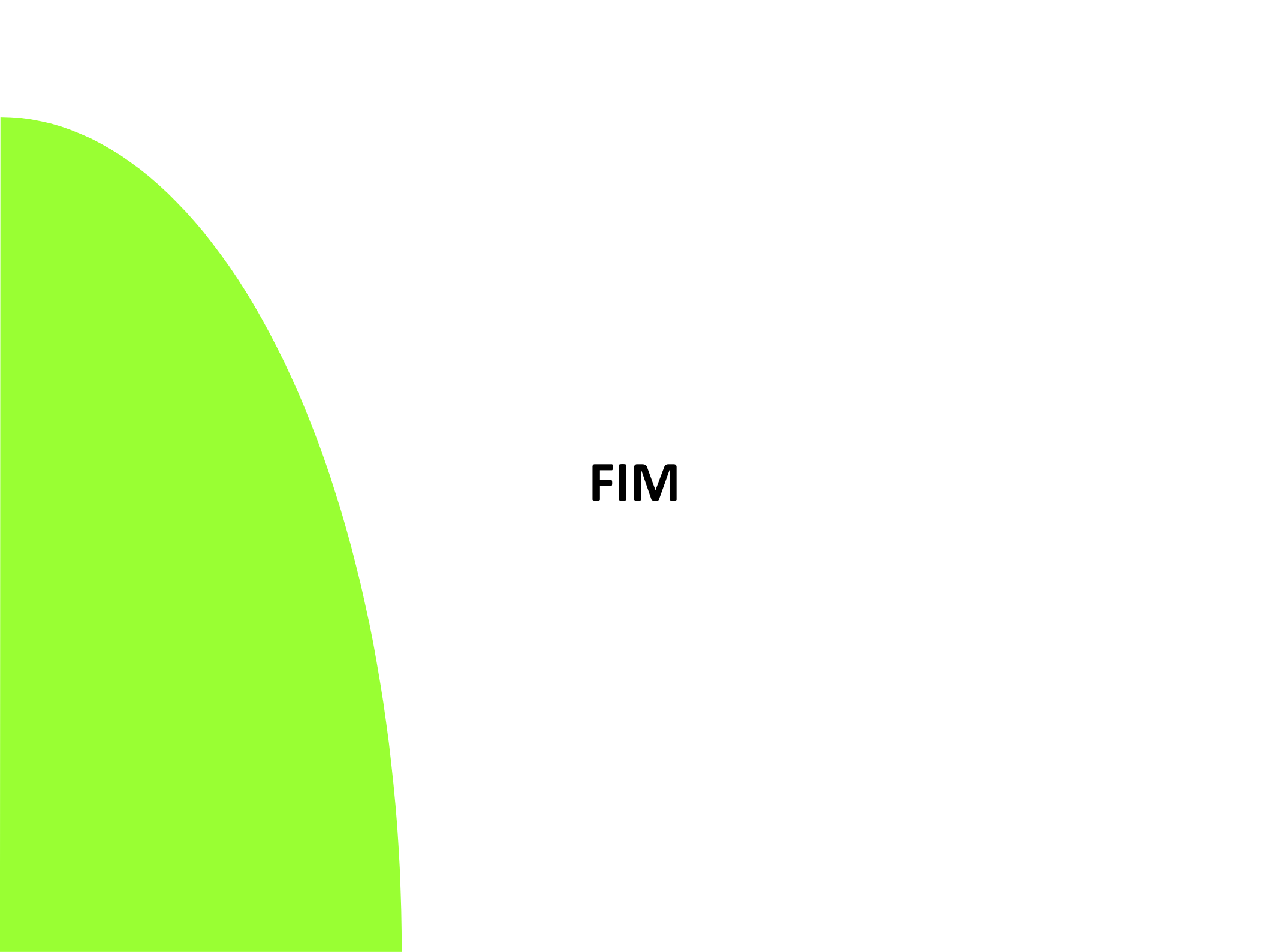
- Exemplo 2) *Trigger* com SIGNAL no MySQL -

```
delimiter //
use test//
create table trigger_test
(
    id int not null
)//
drop trigger if exists trg_trigger_test_ins //
create trigger trg_trigger_test_ins before insert on trigger_test
for each row
begin
    declare msg varchar(128);
    if new.id < 0 then
        set msg = concat('MyTriggerError: Trying to insert a negative value in
trigger_test: ', cast(new.id as char));
        signal sqlstate '45000' set message_text = msg;
    end if;
end
//
delimiter ;
```


Referências



- ❑ Notas de Aula – Prof. Angelo Brayner
- ❑ Sistema de Banco de Dados – Silberschatz
- ❑ <https://www.tutorialspoint.com/How-can-we-use-SIGNAL-statement-with-MySQL-triggers>
- ❑ Manual – MySQL 5.7
- ❑ <http://www.mysqltutorial.org/mysql-signal-resignal/>
- ❑ <http://stackoverflow.com/questions/24/throw-an-error-in-a-mysql-trigger>



FIM