

GATILHOS – (TRIGGERS)

Prof. Ricardo Satoshi

Gatilhos [Triggers]

- Um gatilho [*trigger*] é um tipo de procedimento armazenado que é executado automaticamente quando ocorre algum tipo de alteração numa tabela.

Gatilhos "disparam" quando ocorre uma operação INSERT, UPDATE ou DELETE em uma tabela.

Gatilhos [Triggers]

Sintaxe:

DELIMITER <caractere>

CREATE TRIGGER <nometrigger> <momento>
 <ação> **ON** <tabela>

FOR EACH ROW [BEGIN]

....

[END]

DELIMITER ;

Gatilhos [Triggers]

- Na sintaxe anterior temos **<momento>** que poderá ser AFTER ou BEFORE
- **<ação>** que poderá ser INSERT, UPDATE ou DELETE
- on **<Table>** que é o nome da tabela afetada
- **FOR EACH ROW** – para cada linha

Exemplo de aplicação

Supondo a criação das tabelas a seguir:

```
CREATE TABLE TABFUN (  
  IDFUN INT AUTO_INCREMENT PRIMARY KEY,  
  NOMEFUN VARCHAR(40) NOT NULL,  
  CPF NUMERIC(12),  
  DATAADM DATE,  
  SALARIO DECIMAL(10,2),  
  DEPENDENTES INT DEFAULT 0  
)
```

Exemplo de aplicação (cont.)

```
CREATE TABLE TABLOG (  
  IDFUN INT,  
  USUARIO VARCHAR(40),  
  ACAO VARCHAR(15),  
  DATA DATE,  
  HORA TIME );
```

Criando gatilhos

- Gatilhos são sempre criados vinculados a uma determinada tabela.
- Se a tabela for excluída, todos os gatilhos dela são excluídos como consequência.
- Ao criar um gatilho, você pode especificar em qual(is) operação(ões) ele será acionado: INSERT, UPDATE ou DELETE.

Gatilhos para inserção

- Quando é feita a **inclusão** de uma ou mais linhas na tabela, o MySQL Server cria **uma tabela virtual NEW** que contém as linhas que serão incluídas (mas que ainda não foram).
- Essa tabela tem a mesma estrutura da tabela principal. Você pode usar dados dessa tabela da mesma forma que numa tabela real.

Gatilhos para inserção

- Vamos criar um gatilho, chamado **G_LOG1** que será ativado após uma operação INSERT na tabela TABFUN.
- Digite o seguinte comando:

Gatilhos para inserção (cont.)

```
DELIMITER $  
  
CREATE trigger G_LOG1 AFTER INSERT ON TABFUN  
FOR EACH ROW  
BEGIN  
    INSERT INTO TABLOG  
    SET IDFUN=NEW.IDFUN,  
        USUARIO=USER(),  
        ACAO= 'INSERIU',  
        DATA=CURDATE(),  
        HORA=CURTIME();  
  
END$  
  
DELIMITER ;
```

**Para cada linha inserida em TABFUN uma linha será inserida em TABLOG*

Gatilhos para exclusão

- Na exclusão, as linhas da tabela a serem removidas são colocadas na tabela virtual **OLD** que tem a mesma estrutura da tabela principal.
- Um gatilho para exclusão pode usar OLD para saber as linhas que serão excluídas.

Gatilhos para exclusão

- Vamos criar um gatilho, para a tabela TABFUN para que antes de um funcionário ser excluído esta informação seja registrada em TABLOG.
- Escreva o seguinte:

Gatilho para exclusão (cont.)

```
DELIMITER $  
  
CREATE trigger G_LOG2 BEFORE DELETE ON TABFUN  
FOR EACH ROW  
BEGIN  
    INSERT INTO TABLOG  
    SET IDFUN=OLD.IDFUN,  
        USUARIO=USER(),  
        ACAO='EXCLUIU',  
        DATA=CURDATE(),  
        HORA=CURTIME();  
  
END$  
  
DELIMITER ;
```

**Para cada linha excluída em TABFUN uma linha será inserida em TABLOG*

Gatilhos para atualização

- As tabelas **NEW** e **OLD** como já vimos, são tabelas virtuais que podem ser usadas dentro de um gatilho.
- A primeira contém os dados que estão sendo inseridos na tabela real e a segunda contém os dados antigos, que estão sendo excluídos.

Gatilhos para atualização

- Num gatilho de atualização (**UPDATE**), essas duas tabelas ficam disponíveis.
No caso, **OLD** permite acessar os dados como eram *antes* da modificação e **NEW** permite acessar os dados *depois* da atualização.

Gatilhos para atualização

- Podemos então considerar uma **atualização** como uma ***exclusão*** seguida de uma ***inserção*** (excluem-se valores antigos e inserem-se valores novos).

Gatilhos para atualização (cont.)

- Um gatilho simples para atualização seria:

```
DELIMITER $
```

```
CREATE trigger G_LOG3 BEFORE UPDATE ON TABFUN
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO TABLOG
```

```
    SET IDFUN=NEW.IDFUN,
```

```
        USUARIO=USER(),
```

```
        ACAO= 'ALTEROU' ,
```

```
        DATA=CURDATE(),
```

```
        HORA=CURTIME();
```

```
END$
```

```
DELIMITER ;
```

**Para cada linha atualizada em TABFUN uma linha será inserida em TABLOG*

Gatilhos para atualização

- Para demonstrar melhor o uso das tabelas virtuais OLD e NEW, vamos criar uma nova tabela para auditar os dados de uma atualização:

- Esta tabela seria:

```
CREATE TABLE TABLOGSAL (  
  IDFUN INT,  
  USUARIO VARCHAR(40),  
  DATA DATE,  
  HORA TIME,  
  SALANTES DECIMAL(10,2),  
  SALDEPOIS DECIMAL (10,2) );
```

Gatilhos para atualização (cont.)

- Vamos criar um gatilho em G_LOG4 que vai registrar o salário antes e depois da alteração

DELIMITER \$

```
CREATE trigger G_LOG4 AFTER UPDATE ON TABFUN
FOR EACH ROW
BEGIN
    INSERT INTO TABLOGSAL
    SET IDFUN=NEW.IDFUN,
    USUARIO=USER(),
    DATA=CURDATE(),
    HORA=CURTIME(),
    SALANTES=OLD.SALARIO,
    SALDEPOIS=NEW.SALARIO;
END$
DELIMITER ;
```

Gatilhos - Triggers

- Para se conhecer os gatilhos ativos usa-se
 - **SHOW TRIGGERS**
- Para se conhecer o código de um gatilho específico usa-se
 - **SHOW CREATE TRIGGER <nome_do_tigger>**

Gatilhos – considerações finais

- Apenas um gatilho pode ser usado com um mesmo momento e uma mesma ação sobre uma mesma tabela.
- Assim não é possível criar, por exemplo, 2 gatilhos before insert para uma mesma tabela, sendo entretanto possível criar um before insert e outro after insert para a mesma tabela.
- *OBS:(isso era válido para versões do mysql anteriores a 5.7.2)*