

# TRIGGERS



**Certified MySQL 5.0 Developer**  
**Certified MySQL 5.0 Database Administrator**  
**Certified MySQL 5.1 Cluster Database Administrator**



# Triggers

- Um **TRIGGER** ou gatilho é um objeto de banco de dados, associado a uma tabela, definido para ser disparado, respondendo a um evento em particular;
- Tais eventos são os comandos da **DML** (*Data Manipulation Language*): **INSERT**, **REPLACE**, **DELETE** ou **UPDATE**;
- Podemos definir inúmeros **TRIGGERS** em uma base de dados baseados diretamente em qual dos comandos acima irá dispará-lo, sendo que, para cada um, podemos criar apenas um **TRIGGER**.
- Os **TRIGGERS** poderão ser disparados para trabalharem antes ou depois do evento;

# Triggers

↪ A sintaxe geral de definição de um **TRIGGER** é a seguinte:

```
CREATE [DEFINER = { user | CURRENT_USER }] TRIGGER trigger_name  
trigger_time trigger_event ON tbl_name FOR EACH ROW  
trigger_stmt
```

↪ DEFINER: Quando o **TRIGGER** for disparado, esta opção será checada para checar com quais privilégios este será disparado. Utilizará os privilégios do usuário informado em user ('wagner'@'localhost') ou os privilégios do usuário atual (CURRENT\_USER). Caso essa sentença seja omitida da criação do **TRIGGER**, o valor padrão desta opção é CURRENT\_USER();

↪ trigger\_name: define o nome do procedimento, por exemplo, trg\_test;

# Triggers

- trigger\_time: define se o **TRIGGER** será ativado antes (**BEFORE**) ou depois (**AFTER**) do comando que o disparou;
- trigger\_event: aqui se define qual será o evento, **INSERT**, **REPLACE**, **DELETE** ou **UPDATE**;
- tbl\_name: nome da tabela onde o **TRIGGER** ficará "pendurado" aguardando o trigger\_event;
- trigger\_stmt: as definições do que o o **TRIGGER** deverá fazer quando for disparado.

# Triggers

- ✎ No MySQL, diferentemente de outros **SGBD's**, não é permitido disparar mensagens, por exemplo, “registro atualizado/inserido/excluído com sucesso”;
- ✎ **TRIGGERS** não podem iniciar nem finalizar uma transação;
- ✎ Podem conter mais de um comando em meio à **BEGIN/END**;
- ✎ Podem facilmente acessar a área temporária do **SGBD** para se trabalhar com o valor de “antes” e “depois”;
- ✎ Para cada tabela de um banco de dados, podemos ter no máximo 6 **TRIGGERS**;
- ✎ Os **TRIGGERS** ficam armazenados na tabela **TRIGGERS**, dentro do dicionário de dados;

# Triggers

- ✎ **TRIGGERS** são mais efetivos que restrições check;
- ✎ Podem atuar em uma tabela com base em outra tabela de qualquer banco de dados dentro do MySQL;

# Triggers

## 🐞 Eventos de um **TRIGGER**:

- **BEFORE**: dispara o **TRIGGER** antes dos dados chegarem à tabela;
- **AFTER**: dispara o **TRIGGER** após a atualização dos dados em uma tabela;

```
mysql> create trigger trg_country_up
-> after update
-> on Country
-> for each row
-> begin
->   set @var_old = OLD.Name;
->   set @var_new = NEW.Name;
-> end;
-> //
```

Query OK, 0 rows affected (0.00 sec)

# Triggers

## OLD e NEW

- **OLD**: operador para ser utilizado no interior da **TRIGGER** para recuperar o valor anterior à um **UPDATE**, **\*REPLACE** ou **DELETE**;
- **NEW**: operador para ser utilizado no interior da **TRIGGER** para recuperar dados atuais a partir de comandos **INSERT**, **\*REPLACE** e **UPDATE**.



# Exercício

Artigo: [MySQL - TRIGGERS](#)

Com base no assunto apresentando “*Triggers*”, responda a **LISTA 10**.



Fonte de referência: <http://dev.mysql.com/doc/refman/5.0/en/create-trigger.html>