TRIGGERS NO MYSQL

Triggers Definição

- Assim como um gatilho dispara um projétil de uma arma para que este possa acertar o seu alvo, um *trigger* dispara uma série de ações para cumprir um objetivo específico.

- Trigger é uma instrução SQL (ou um conjunto de instruções SQL) que fica armazenado no banco de dados e será ativado ou disparado quando um evento ocorrer numa tabela.

Triggers Definição

- Um Trigger é automaticamente executado quando um comando do tipo **INSERT**, **DELETE** ou **UPDATE** é executado em uma tabela.
- Um trigger é uma regra do tipo E_C_A:
  - E: Evento
  - C: Condição a ser satisfeita na presença do evento E
  - A: Ação a ser tomada caso a condição C seja satisfeita

Vínculo de um trigger

- Os gatilhos (triggers) são sempre vinculados a uma determinada tabela
- Quando uma tabela é removida, todos os gatilhos relacionados serão excluídos automaticamente.

Usos e Aplicações

- Impor uma integridade de dados mais complexa do que uma restrição CHECK;
- Definir mensagens de erro personalizadas;
- Comparar a consistência dos dados – posterior e anterior – de uma instrução UPDATE;
- Os TRIGGERS são usados com enorme eficiência para impor e manter integridade referencial de baixo nível, e não para retornar resultados de consultas.

Usos e Aplicações

- TRIGGERS podem ser usados para atualizações e exclusões em cascata através de tabelas relacionadas em um banco de dados.

Limitações dos Triggers

- Não podem utilizar SHOW, DATA, LOAD TABLE, BACKUP DATABASE, RESTORE, FLUSH e RETURN

- Podem utilizar COMMIT, ROLLBACK, START TRANSACTION, LOCK/UNLOCK TABLES, ALTER, CREATE, DROP, RENAME

- Podem utilizar PREPARE, EXECUTE

- Podem chamar stored procedures ou funções

Sintaxe

CREATE TRIGGER nome tempo evento

ON tabela

FOR EACH ROW

BEGIN

…

…

END

Exemplo:

- **Trigger Backup**

Criar tabela de clientes (Tb_cliente):

```
Create table Tb_clientes
Cod_cli  integer not null,
Nome_cli varchar(15),
Sexo_cli varchar(2),
Primary key (cod_cli));
```

- **Criar tabela que irá receber copia dos dados (Tb_copiaclientes):**

```
Create table Tb_copiaclientes
Código_cop integer not null,
Nome_cop varchar(15),
Sexo_cop varchar(2),
Primary key (código_cop));
```

- **Criar trigger que irá realizar backup de inserção de cliente (copiacliente):**

```
Create trigger copiacliente after  insert
on Tb_clientes
For each row
Insert into tb_copiaclientes (codigo_cop,
  nome_cop, sexo_cop)
Values(new.cod_cli, new.nome_cli,
  new.sexo_cli);
```

- **Testar funcionamento da trigger:**

Insert into tb_clientes
Values (1, 'Antonio','M');

Select * from tb_clientes;
Resultado: 1 | Antonio | M

Select * from tb_copiaclientes
Resultado: : 1 | Antonio | M

Dois comandos SQL relacionados a TRIGGERS:
"Show triggers from nome_do_banco"
"Drop trigger trigger_name"