

# BANCO DE DADOS



## TRIGGER

Prof: Ricardo Rufino

# Trigger: Introdução



- 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 definir apenas um TRIGGER. Os TRIGGERS poderão ser disparados para trabalharem antes ou depois do evento.

# Trigger: Introdução



- A tabela abaixo será utilizada como base dos exemplos

```
CREATE TABLE tbl_cliente(  
    cliente_id int auto_increment primary key,  
    cliente_nome varchar(80) not null,  
    cliente_email varchar(80) not null  
);
```

# Trigger: Introdução

- Baseado na tabela **tbl\_cliente**, podemos definir os TRIGGERS para serem disparados, por exemplo, antes (BEFORE) ou depois (AFTER) de um INSERT. Agora sabemos então que para cada momento BEFORE ou AFTER, podemos ter um TRIGGER a ser disparado para defender alguma lógica.

# Trigger: Sintaxe

- 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
```

# Trigger: Sintaxe

- **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;
- **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.

# Definições: OLD - NEW

- Em meio aos TRIGGERS temos dois operadores importantíssimos que nos possibilitam acessar as colunas da tabela alvo do comando DML, ou seja, podemos acessar os valores que serão enviados para a tabela `tbl_cliente` antes (BEFORE) ou depois (AFTER) de um UPDATE, por exemplo. Tais operadores nos permitirão então, ter dois momentos, o antes e o depois e também examinar os valores para que sejam ou não inseridos, atualizados ou excluídos da tabela.

# Diretrizes: OLD - NEW

- INSERT: o operador NEW.nome\_coluna, nos permite verificar o valor enviado para ser inserido em uma coluna de uma tabela. OLD.nome\_coluna não está disponível.
- DELETE: o operador OLD.nome\_coluna nos permite verificar o valor excluído ou a ser excluído. NEW.nome\_coluna não está disponível.
- UPDATE: tanto OLD.nome\_coluna quanto NEW.nome\_coluna estão disponíveis, antes (BEFORE) ou depois (AFTER) da atualização de uma linha.



# Exemplo

- O Trigger fará nada mais que validar se os dados foram passados em uma declaração INSERT antes (BEFORE) que sejam cadastrados na tabela de exemplo. Validaremos o nome com quantidade de caracteres maior ou igual a 4 (quatro)

```
CREATE TRIGGER tgr_1 BEFORE INSERT ON tbl_cliente
FOR EACH ROW
BEGIN
    SET @nome = NEW.cliente_nome;
    IF((CHAR_LENGTH(@nome) <=4 ) OR (@nome="")) then
        set NEW.cliente_nome = NULL;
    END IF;
END
```

# Exercícios

