SENAI

Faculdade de Tecnologia SENAI "Roberto Mange"

Aula 9 – Triggers

Triggers SQL

O Trigger no MySQL é um objeto de banco de dados (programa) associado a uma tabela. O termo trigger (gatilho em inglês) define uma estrutura do banco de dados que funciona, como o nome sugere, como uma função que é disparada mediante alguma ação. Será ativado quando uma ação definida for executada para a tabela. O trigger pode ser executado quando você executa uma das seguintes instruções do MySQL na tabela: INSERT , UPDATE e DELETE e pode ser chamada antes ou depois do evento

Principais triggers

Existem dois tipos principais: o AFTER e o BEFORE.

O AFTER é executado depois do evento que disparou o trigger. Ao passo que o BEFORE executa o trigger antes da ação DML.

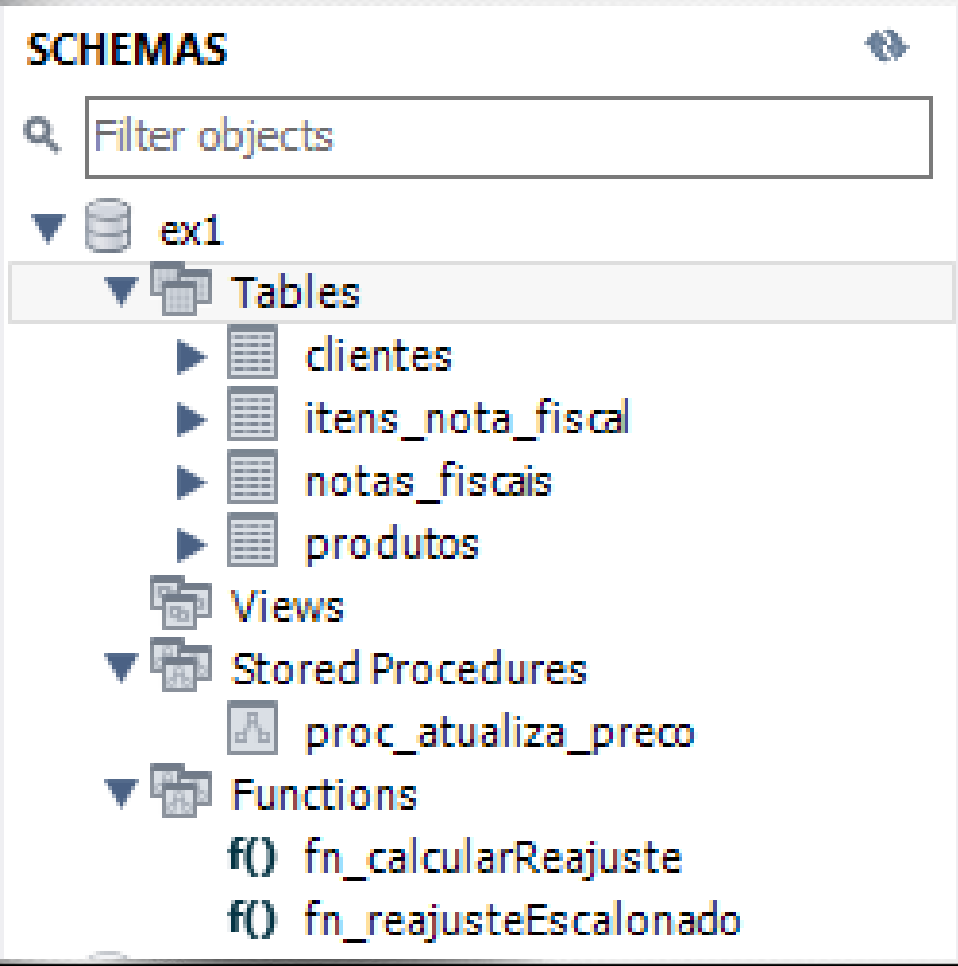
Principais triggers

Triggers de linha ➔ executado sempre que um linha é modificada, deverá ser acompanhada pela instrução “for each row”. Toda vez que a linha for modificada o trigger será executado também.

Triggers de instrução ➔ executado uma única vez através de um comando DML,

Triggers de Logon ➔ há ainda um que é executado quando um determinado usuário se conecta ao BD, não é comum a todos os BD dependendo do seu SGBD

Exemplo 1 - Triggers

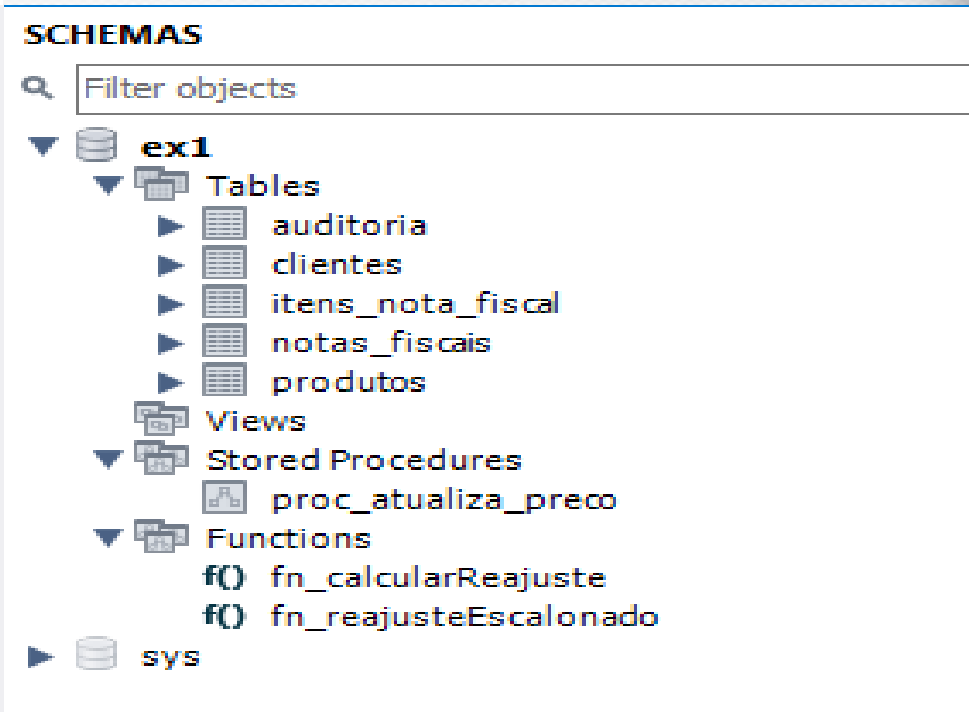


Para esse exemplo vamos usar o BD do exercício 1 que estamos habituado. Vamos criar uma tabela chamada auditoria que através de um trigger vai gerar um histórico de alteração em alguma dessas tabelas. Podemos novamente usar a tabela de produtos.

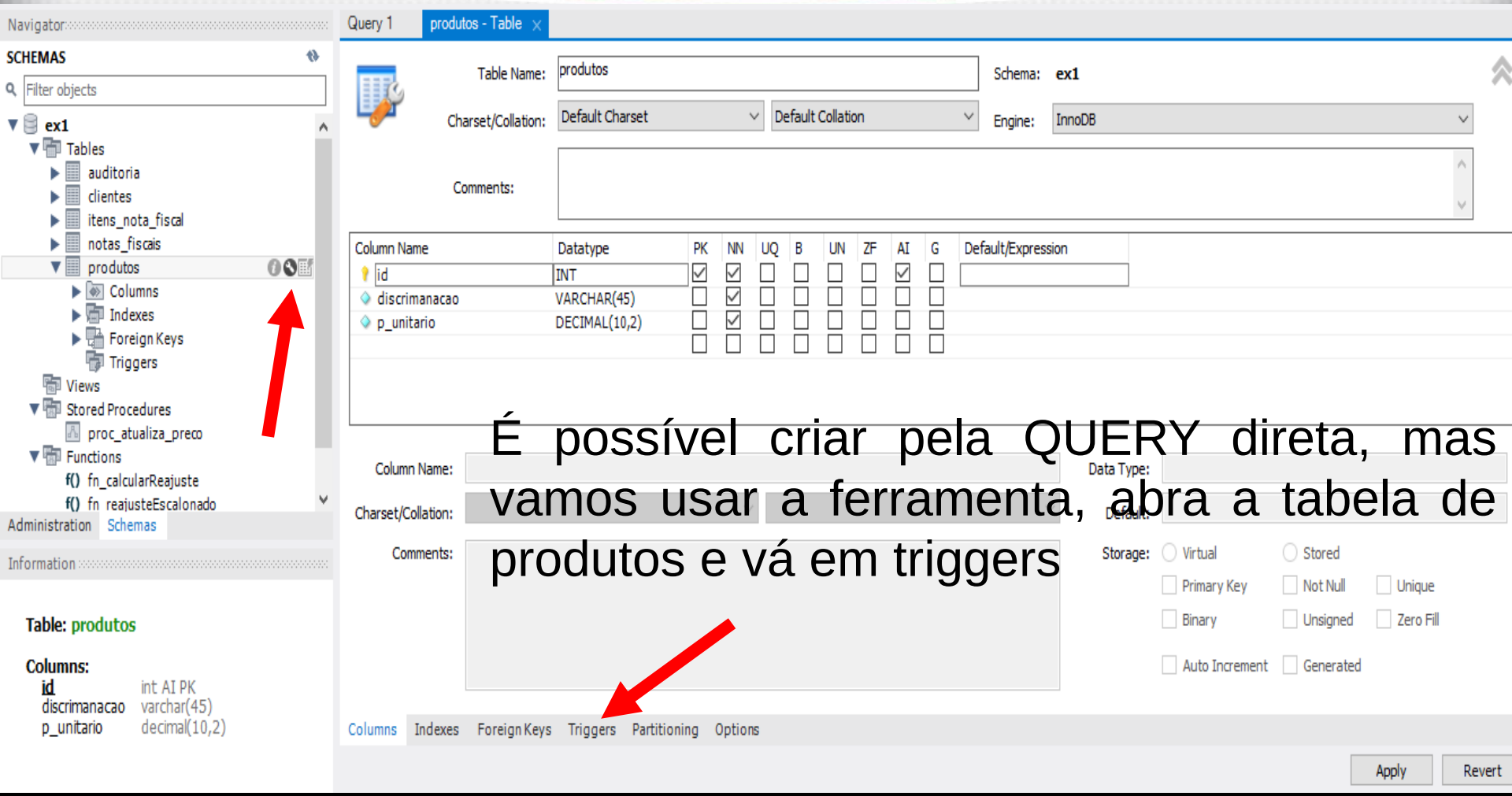
Exemplo 1 - Triggers

```
-- Criando uma nova tabela para demonstrar a ação do trigger

CREATE TABLE auditoria(

discriminacao VARCHAR(50),

p_unitario_antigo DECIMAL(10,2),

p_unitario_novo DECIMAL(10,2),

data_trigger DATE

);
```

Crie uma nova tabela para poder lançar nela as ações da execução do trigger

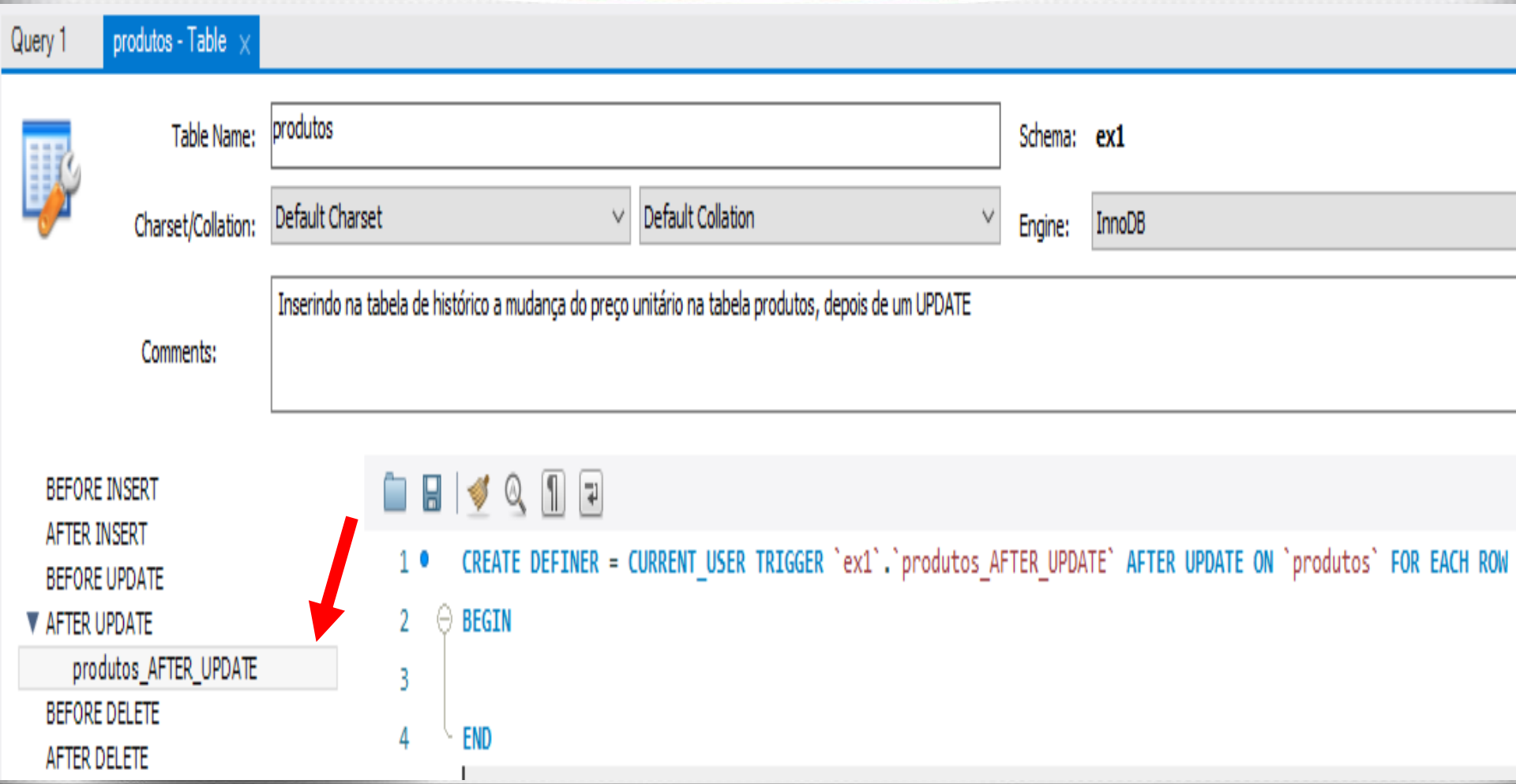


Exemplo 1 - Triggers



É possível criar pela QUERY direta, mas vamos usar a ferramenta, abra a tabela de produtos e vá em triggers

Exemplo 1 - Triggers



Exemplo 1 - Triggers

```
1  CREATE DEFINER=`root`@`localhost` TRIGGER `produtos_AFTER_UPDATE` AFTER UPDATE ON `produtos` FOR EACH ROW
2  BEGIN
3      INSERT INTO auditoria -- INSERE NA TABELA AUDITORIA
4      VALUE (OLD.discrimanacao  , OLD.p_unitario, NEW.p_unitario, curdate());
5      -- OS VALORES ANTES DO UPDATE e OS VALORES DEPOIS DO UPDATE
6      -- Os valores antigos serão capturados pela palavra chave OLD
7      -- Os valores novos serão capturados pela palavra chave NEW
8  END
```

Digite o código a seguir e dê apply. Veja que foi inserido na tabela produtos esse trigger

Testando o exemplo 1

```
SELECT *FROM ex1.auditoria;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

discriminacao	p_unitario_antigo	p_unitario_novo	data_trigger

```
SELECT *FROM ex1.produtos;
```

Result Grid | Filter Rows:

id	discrimanacao	p_unitario
1	Notebool	8318.75
2	Laptop	9558.99
3	MEMÓRIA RAM	1068.66
4	Celular SAMSUNG	2778.59
NULL	NULL	NULL

Testando o exemplo 1

```
UPDATE produtos SET p_unitario = 8999.99 WHERE id = 1;
```

Result Grid | Filter Rows: | Edit:

id	discrimanacao	p_unitario
1	Notebool	8999.99
2	Laptop	9558.99
3	MEMÓRIA RAM	1068.66
4	Celular SAMSUNG	2778.59
NULL	NULL	NULL

```
SELECT *from auditoria;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

discriminacao	p_unitario_antigo	p_unitario_novo	data_trigger
Notebool	8318.76	8999.99	2023-10-30