SENAI

Faculdade de Tecnologia SENAI "Roberto Mange"

Aula 8 – Procedures e Funções

Procedimentos SQL

Segundo Silberschatz (2010), os procedimentos armazenados (Stored procedures) foram definidos na versão SQL 1999, para que fosse obtida capacidade procedural (ações estruturadas) nos bancos de dados, porém, não com a intenção de substituir técnicas que também estão disponíveis nas linguagens de programação, como Java, C++ ou C#. Os procedimentos não permitem o retorno de sua ação, sendo basicamente a principal diferenciação das funções que veremos posteriormente.

Procedimentos SQL

Esse recurso deve permitir armazenar procedimentos como seleção de dados, exclusão de registros, alteração de dados, entre outras funções disponíveis na linguagem de programação de banco de dados SQL.

Procedimentos SQL

```
CREATE PROCEDURE nome_do_procedure (var_nome
tipo)
   Declarações.
```

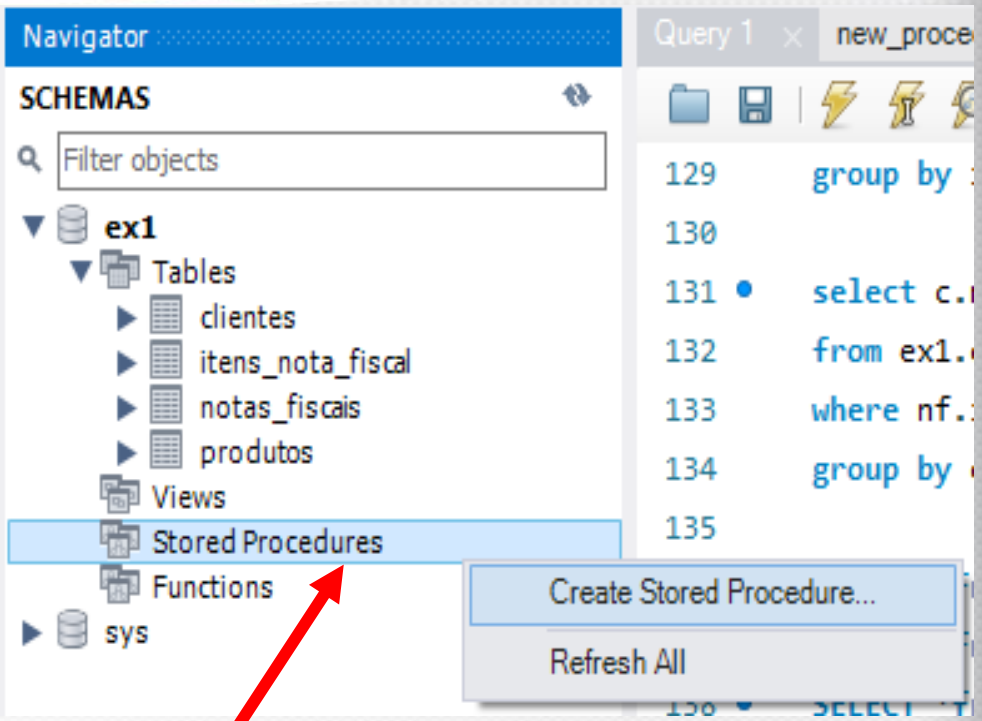
- nome_da_procedure : identifica o nome do procedimento. Como boa prática utilize o prefixo proc. Por exemplo proc_teste.
- (var_nome tipo): deve ser criada uma variável. Uma boa prática é utilizar o prefixo var e, em seguida, o tipo dessa variável. Por exemplo: var_teste int;
- Declarações: podem ser utilizadas as seleções de dados

Exemplo Procedimentos SQL

BASE DE DADOS USADA EX1

```
SELECT *from produtos;
```

	id	discrimanacao	p_unitario
▶	1	Notebool	5999.99
	2	Laptop	7899.99
	3	MEMÓRIA RAM	485.75
	4	Celular SAMSUNG	2525.99
*	NULL	NULL	NULL



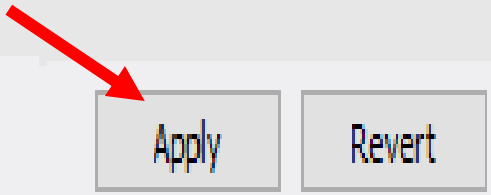
Esse SGBD tem uma ferramenta para auxiliar a criação de stored procedures, clique com o botão direito e peça para criar um procedimento

Exemplo Procedimentos SQL

```
1  CREATE PROCEDURE `proc_atualiza_preco`(var_novoValor decimal(10,2), var_nomeProduto varchar(45))
2  BEGIN
3      UPDATE produtos set p_unitario = var_novoValor where discrimanacao = var_nomeProduto;
4      SELECT *FROM produtos;
5  END
6
```

Routine

Vamos fazer um procedimento para alterar preços unitários, dependendo do nome ou discriminação do produto. Clique em apply que a ferramenta fará alguns ajustes e dê apply de novo.



Exemplo Procedimentos SQL

```
CALL proc_atualiza_preco(6250.00, "Notebool");
```

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

	id	discrimanacao	p_unitario
▶	1	Notebool	6250.00
	2	Laptop	7899.99
	3	MEMÓRIA RAM	485.75
	4	Celular SAMSUNG	2525.99

Funções SQL

Segundo Silberschatz (2010), as funções são definidas na linguagem SQL a partir da versão SQL:2003. Essa técnica possibilita realizar cálculos aritméticos complexos utilizando os valores das colunas existentes em um banco de dados. Basicamente, o motivo de se utilizar uma FUNCTION é retornar tabelas como resultado, conhecidas como funções de tabela.

Funções SQL

```
CREATE FUNCTION nome_da_funcao (x tipo, y tipo)
RETURNS tipo
RETURN (função);
```

- nome_da_função: pode ser escolhido pelo desenvolvedor. Uma boa prática e nomeá-las com o prefixo fn, por exemplo fn_teste;
- (x tipo, y tipo): são declaradas duas variáveis (x e y), e os seus respectivos tipos. Por exemplo: (w int, z decimal(6,2)).

Funções SQL

```
CREATE FUNCTION nome_da_funcao (x tipo, y
tipo)
RETURNS tipo
RETURN (função);
```

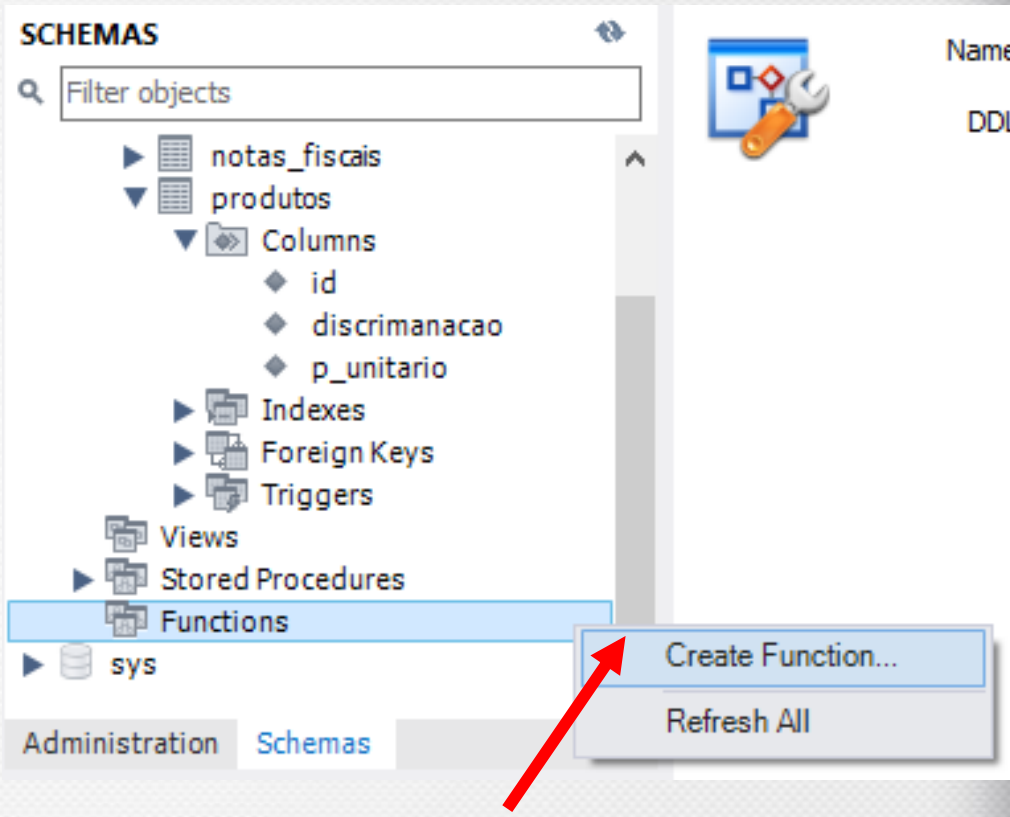
- RETURNS tipo: determina que tipo de dado será retornado após a execução da função. Por exemplo: RETURNS decimal(6,2).
- RETURN (função): e o trecho da expressão em que são definidas as expressões aritméticas, determinadas em (x tipo, y tipo). Por exemplo: RETURN (x + 2) – (y + 1).

Exemplo 1
Funções SQL

BASE DE DADOS USADA EX1

Result Grid | Filter Rows:

	id	discrimanacao	p_unitario
▶	1	Notebool	6250.00
	2	Laptop	7899.99
	3	MEMÓRIA RAM	485.75
	4	Celular SAMSUNG	2525.99
*	NULL	NULL	NULL



Use a ferramenta de auxílio do SGBD para criar a função

Exemplo 1
Funções SQL

fn_calcularReajuste

The name of th
statement. The

```
1  CREATE FUNCTION `fn_calcularReajuste`(codigoProduto INT, percentual INT)
2  RETURNS decimal(10,2)
3  BEGIN
4      DECLARE precoAtual, resultado DECIMAL(10,2);
5      SELECT p_unitario INTO precoAtual FROM produtos WHERE id = codigoProduto;
6      SET resultado = precoAtual * (1 + percentual/100);
7  RETURN resultado;
8  END;
9
10
11
12
```

Crie a rotina a seguir para calcular o aumento de um produto e clique em apply

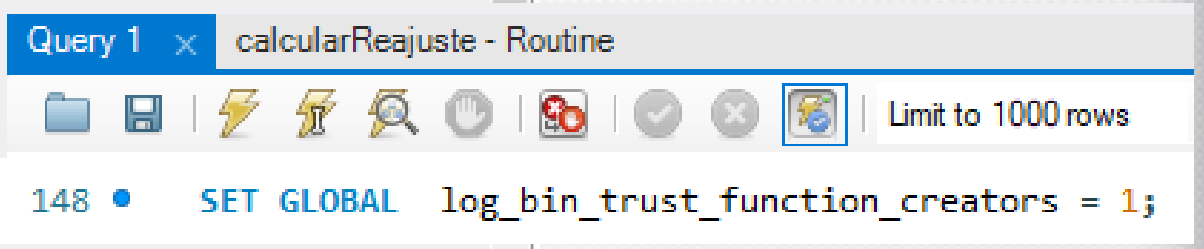
Exemplo 1 Funções SQL

Message Log

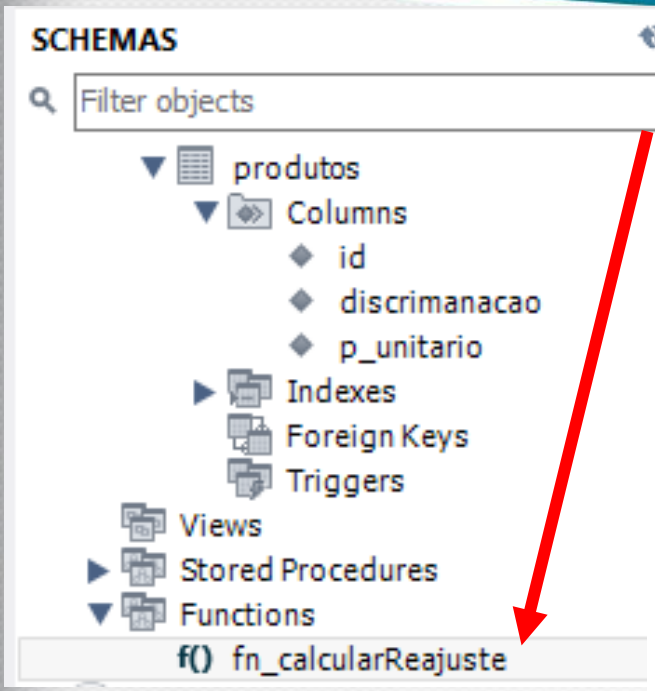
```
   WHERE id = codigoProduto;
RETURN precoNovo = precoNovo * (percentual1 /100);
END;$$

DELIMITER ;

Operation failed: There was an error while applying the SQL script to the database.
ERROR 1418: This function has none of DETERMINISTIC, NO SQL, or READS SQL DATA in its declaration
and binary logging is enabled (you *might* want to use the less safe log_bin_trust_function_creators
variable)
SQL Statement:
CREATE FUNCTION `calcularReajuste` (percentual1 INT, codigoProduto INT)
RETURNS NUMERIC
BEGIN
        DECLARE precoNovo NUMERIC;
        SELECT p_unitario
   INTO precoNovo
   FROM produtos
   WHERE id = codigoProduto;
RETURN precoNovo = precoNovo * (percentual1 /100);
END;
```

Se quando ela for criada indicar que ela precisa ser determinística, rode a parametrização a seguir na query ➔

SET GLOBAL log_bin_trust_function_creators = 1;



Exemplo 1
Funções SQL



Diferente dos procedimentos a função retorna algo e portanto podemos usá-la nas sintaxes comuns SQLs usadas.

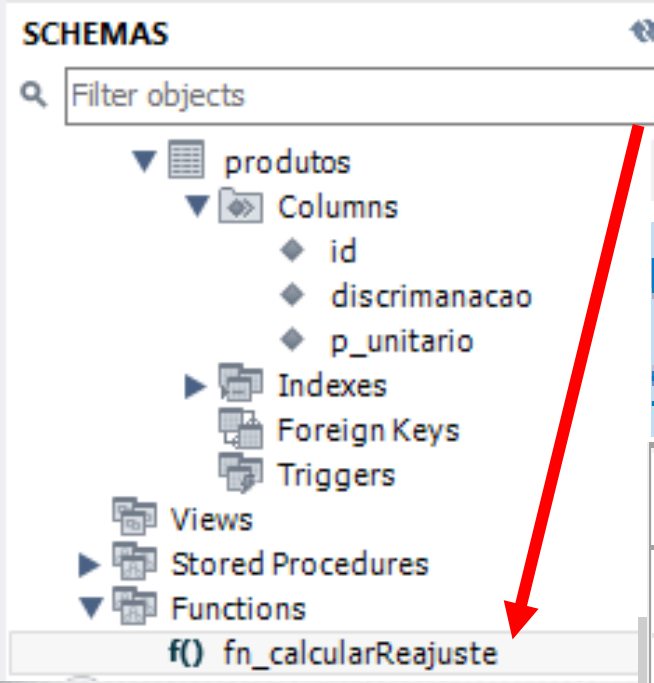
```
SELECT fn_calcularReajuste(1,10) AS precoNovo; -- imprimindo o novo preço
```

precoNovo
6875.00

```
UPDATE produtos SET p_unitario = fn_calcularReajuste(1,10) where id=1; -- alterando um unico item

SELECT *from produtos;
```

id	discrimanacao	p_unitari
1	Notebool	6875.00
2	Laptop	7899.99
3	MEMÓRIA RAM	485.75
4	Celular SAMSUNG	2525.99
NULL	NULL	NULL

Exemplo 1
Funções SQL



Diferente dos procedimentos a função retorna algo e portanto podemos usá-la nas sintaxes comuns SQLs usadas.

```
UPDATE produtos SET p_unitario = fn_calcularReajuste(id,10) ; -- alterando todos os itens da lista

SELECT *from produtos;
```

id	discrimanacao	p_unitario
1	Notebool	7562.50
2	Laptop	8689.99
3	MEMÓRIA RAM	534.33
4	Celular SAMSUNG	2778.59
NULL	NULL	NULL

Exemplo 2
Funções SQL

Name: fn_reajusteEscalonado

The name of the ro
statement. The DD

DDL:

```
1  CREATE FUNCTION `fn_reajusteEscalonado`(codigoProduto INT, perc1 INT, perc2 INT)
2  RETURNS decimal(10,2)
3  BEGIN
4      DECLARE precoAtual, resultado DECIMAL(10,2);
5      SELECT p_unitario INTO precoAtual FROM produtos WHERE id = codigoProduto;
6      IF precoAtual <= 2000 THEN
7          SET resultado = precoAtual * (1 + perc1/100);
8      ELSEIF precoAtual > 2000 AND precoAtual < 5000 THEN
9          SET resultado = precoAtual;
10     ELSEIF precoAtual >= 5000 THEN
11         SET resultado = precoAtual * (1 + perc2/100);
12     END IF;
13 RETURN resultado;
14 END;
```

Criando outra função para selecionar um cálculo de reajuste personalizado, com condicionais

Exemplo 2
Funções SQL

```
UPDATE produtos SET p_unitario = fn_reajusteEscalonado (id,100,10) ;
SELECT *from produtos;
```

	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

Exercício

1) Usando o banco de dados da nota fiscal, criar uma função para chamar o valor total dela e aplicar um desconto progressivo:
- Abaixo ou igual a 1000 sem desconto;
- Acima de 1000 ou menor e igual a 10000 com 10% desconto;
- Acima de 10000 com 20% de desconto

2) Gerar uma consulta do nota fiscal com procedimentos, com item, produto, quantidade, valor unitário, sub total, desconto e valor total com desconto. Passando como parâmetro a id da nota fiscal