



Banco de Dados

13 – Procedures e Functions

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br



Stored Procedures e Functions

Stored Procedures e Functions

Uma **Stored Procedure** e **Function** são um conjunto de comandos SQL que podem ser armazenados no servidor. Uma vez que isto tenha sido feito, os clientes não precisam de reenviar os comandos individuais mas podem fazer referência.

Stored Procedure e **Function** podem fornecer um aumento no desempenho já que menos informação precisa ser enviada entre o servidor e o cliente.

O lado negativo é que isto aumenta a carga no sistema do servidor de banco de dados, já que a maior parte do trabalho é feita no servidor e menor parte é feita do lado do cliente.

Stored Procedures e Functions

Criação de Stored Procedure:

```
DELIMITER $$  
CREATE PROCEDURE nome_procedure (parametro_1 , ..., OUT parametro_n)  
BEGIN  
  
END $$  
DELIMITER ;
```

Exemplo:

```
DELIMITER $$  
CREATE PROCEDURE sp_soma (Numero_1 int, Numero_2 int, OUT result int)  
BEGIN  
    set result = Numero_1 + Numero_2  
END $$  
DELIMITER ;
```

Stored Procedures e Functions

Criação de Function:

```
DELIMITER $$  
CREATE FUNCTION nome_função(parametro_1,...,parametro_n) RETURNS tipo  
BEGIN  
    return nome_variavel;  
END $$  
DELIMITER ;
```

Exemplo:

```
DELIMITER $$  
CREATE FUNCTION fn_soma (Numero_1 int, Numero_2 int) RETURNS int  
BEGIN  
    declare resultado;  
    set resultado = Numero_1 + Numero_2  
    return resultado;  
END $$  
DELIMITER ;
```

Stored Procedures e Functions

Declaração de variável:

```
declare nome_variavel tipo_variavel;
```

Atribuir valor a variável:

```
Set nome_variavel = valor
```

```
SELECT      campo_tabela  
INTO        nome_variavel  
FROM        tabela  
WHERE       condição
```

Stored Procedures e Functions

Alteração de Stored Procedure:

```
DELIMITER $$
```

```
ALTER PROCEDURE nome_procedure (param_1, . . . , param_n OUT)
```

```
BEGIN
```

```
END $$
```

```
DELIMITER ;
```

Stored Procedures e Functions

Alteração de Function:

```
DELIMITER $$
```

```
ALTER FUNCTION nome_função (param_1,...,param_n) RETURNS tipo  
BEGIN
```

```
    return nome_variavel;
```

```
END $$
```

```
DELIMITER ;
```


Stored Procedures e Functions

Remoção de Stored Procedure:

```
DROP PROCEDURE nome_procedure;
```

Remoção de Function:

```
DROP FUNCTION nome_função;
```

Stored Procedures e Functions

Visualizar o conteúdo de uma Stored Procedure:

```
SHOW CREATE PROCEDURE nome_procedure;
```

Visualizar o conteúdo de uma Function:

```
SHOW CREATE FUNCTION nome_função;
```

Stored Procedures e Functions

Utilizar uma Stored Procedure:

```
CALL nome_procedure (param_1, ..., param_n OUT);
```

Utilizar uma Function:

```
SELECT nome_função (param_1, ..., param_n);
```

Stored Procedures e Functions

Recursos dentro Stored Procedure:

IF

```
IF condição THEN
    bloco_codigo
ELSEIF condição THEN
    bloco_codigo
ELSE
    bloco_codigo
END IF
```

Stored Procedures e Functions

Recursos dentro Stored Procedure:

WHILE

```
WHILE condicao DO  
    bloco_codigo  
END WHILE
```

Stored Procedures e Functions

Recursos dentro Stored Procedure:

CASE

```
CASE variável
    WHEN valor THEN bloco_codigo
    WHEN valor THEN bloco_codigo
    ELSE bloco_codigo
END CASE
```

Stored Procedures e Functions

Recursos dentro Stored Procedure:

REPEAT

REPEAT

 bloco_codigo

 UNTIL condicao

END REPEAT

Stored Procedures e Functions

Recursos dentro Stored Procedure:

Loop

```
variavel_loop LOOP
```

```
    bloco_codigo
```

```
END LOOP
```

Encerra o loop:

```
LEAVE variavel_loop;
```

Faz mais uma interação no loop:

```
ITERATE variavel_loop;
```


Stored Procedures e Functions

Recursos dentro Stored Procedure:

CURSOR

```
DECLARE controle INT DEFAULT 0;
```

```
DECLARE nome_cursor CURSOR FOR consulta;
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET controle = 1;
```

```
OPEN nome_cursor
```

```
FETCH nome_cursor INTO variavel
```

```
CLOSE nome_curso;
```

Stored Procedures e Functions

Recursos dentro Stored Procedure:

CURSOR

```
CREATE PROCEDURE soma_idade(OUT SOMA INT)
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE vidade int;
    DECLARE cur_aluno CURSOR FOR SELECT idade FROM aluno;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    SET SOMA = 0;
    OPEN cur_aluno;
    FETCH cur_aluno INTO vidade;
    while not (done = 1) do
        IF vidade is not null THEN
            set soma = soma + vidade;
        END IF;
        FETCH cur_aluno INTO vidade;
    END while;
    CLOSE cur_aluno;
END !!
```