Banco de Dados II

Técnicas de Programação em Banco de Dados

Procedures – Parte II

Prof. Tadeu Classe
tadeu.classe@uniriotec.br

Procedimentos com outros tipos de Query SQL

- Até o momento vimos procedimentos se comportando como Views, porém os procedimentos vão além do comportamento de exibição como as viwes;

- É possível realizar dentro de procedimentos operações como exclusão, inclusão, alterações e etc.

Procedimentos com outros tipos de Query SQL

- Exercitando:
  - Procedimento para incluir um novo departamento:
    - 1. Crie o SQL para isso:

```
INSERT into departments (dept_no, dept_name)
Values ('d010', 'Informática')
```

Procedimentos com outros tipos de Query SQL

- Exercitando:
  - Procedimento para incluir um novo departamento:
    - 2. Crie o procedimento:

```
DELIMITER $$
CREATE procedure addDept
(deptNum char(4), depNome varchar(50))
BEGIN
    INSERT into departments (dept_no, dept_name)
    Values (deptNum, depNome);
END$$
```

Procedimentos com outros tipos de Query SQL

- Exercitando:
  - Procedimento para incluir um novo departamento:
    - 3. Uso do procedimento:

```
CALL addDept('d011', 'Redes de Computadores');
```

Procedimentos com outros tipos de Query SQL

- Exercitando:
  - Procedimento para incluir um novo departamento:
    - 4. Verifique se realmente funcionou:

```
Select * from departments
```

dept_no	dept_name
d004	Production
d005	Development
d006	Quality Management
d007	Sales
d008	Research
d009	Customer Service
d010	Informática
d011	Redes de Computadores
NULL	NULL

Procedimentos com outros tipos de Query SQL

- Exercitando:
  - Crie um procedimento para incluir novos empregados;

Usando Variáveis

- Declarar variáveis no MySQL é simples:
- Basta utilizar a Sintaxe:

```
SET @<nome variavel> := <valor variavel>
```

- Exemplo:

```
SET @idade := 25;
Select @Idade minhaIdade;
```

Usando Variáveis

- Em procedimentos as variáveis devem ser “declaradas” através do comando DECLARE.

- Exemplo de um procedimento com variáveis:

```
DELIMITER $$
CREATE PROCEDURE nome_idade()
BEGIN

    DECLARE nome varchar(50);
    DECLARE idade int;

    SET nome = 'Tadeu Classe ';
    SET idade = 25;

    Select nome, idade;

END$$
```

Usando Variáveis

- Em procedimentos as variáveis devem ser “declaradas” através do comando DECLARE.

- Exemplo de um procedimento com variáveis:

```
CALL nome_idade();
```

nome	idade
Tadeu Classe	25

Usando Variáveis

- Melhorando o procedimento addDept:

```
DELIMITER $$
CREATE procedure addDept
(depNome varchar(50))
BEGIN

    DECLARE qtdeDept int;

    SELECT count(*) + 1
    INTO qtdeDept
    FROM departments;

    INSERT INTO departments(dept_no, dept_name)
    VALUES (concat('d', qtdeDept), depNome);

END$$
```

Usando Variáveis

- Melhorando o procedimento addDept:

```
DELIMITER $$
CREATE procedure addDept
(depNome varchar(50))
BEGIN

    DECLARE qtdeDept int;

    SELECT count(*) + 1
    INTO qtdeDept
    FROM departments;

    INSERT INTO departments(dept_no, dept_name)
    VALUES (concat('d', qtdeDept), depNome);

END$$
```

Removido o parâmetro de número do departamento;

Usando Variáveis

- Melhorando o procedimento addDept:

```
DELIMITER $$
CREATE procedure addDept
(depNome varchar(50))
BEGIN

    DECLARE qtdeDept int;

    SELECT count(*) + 1
    INTO qtdeDept
    FROM departments;

    INSERT INTO departments(dept_no, dept_name)
    VALUES (concat('d', qtdeDept), depNome);

END$$
```

Declaração da variável de quantidade de departamentos;

Usando Variáveis

- Melhorando o procedimento addDept:

```
DELIMITER $$
CREATE procedure addDept
(depNome varchar(50))
BEGIN

    DECLARE qtdeDept int;

    SELECT count(*) + 1
    INTO qtdeDept
    FROM departments;

    INSERT INTO departments(dept_no, dept_name)
    VALUES (concat('d', qtdeDept), depNome);

END$$
```

Busca da quantidade de departamentos + 1. Sendo colocado o valor na variável criada.

Usando Variáveis

- Melhorando o procedimento addDept:

```
DELIMITER $$
CREATE procedure addDept
(depNome varchar(50))
BEGIN

    DECLARE qtdeDept int;

    SELECT count(*) + 1
    INTO qtdeDept
    FROM departments;

    INSERT INTO departments(dept_no, dept_name)
    VALUES (concat('d', qtdeDept), depNome);

END$$
```

Inclusão do código do departamento concatenado ao seu prefixo.

Usando Variáveis

- Exercícios (Utilizando variáveis):
  - Crie um procedimento que passe por parâmetros os dados para a inclusão de um novo funcionário. Crie variável para controlar o auto incremento do seu campo primário. Inclua no banco de dados um novo registro;
  - Crie um procedimento que passe por parâmetro o nome do departamento e do funcionário. Crie variáveis que receba o código deste funcionário e deste departamento. Inclua um novo registro na tabela dept_emp;

Usando Variáveis

- Exercícios (Utilizando variáveis):
  - Crie um procedimento que passe o nome de um funcionário, se seu título por parâmetros. Recupere em uma variável seu código. Inclua o registro na tabela titles.
  - Crie um procedimento que passe o nome de um funcionário por parâmetro. Recupere em uma variável seu código e inclua o registro na tabela salaries.

Estruturas de Procedimentos

- Os procedimentos são estruturas extremamente poderosas que facilitam a interação dos usuários com o servidor de banco de dados, além de ajudar no controle de acesso aos dados, aumentando a segurança do sistema, como mencionado anteriormente.

- Um dos requisitos que contribuem para essas características dos procedimentos está na possibilidade de uso de estruturas de controle de fluxo de dados, como estruturas de decisão e de repetição.

Estruturas de Procedimentos

- As estruturas de controle de fluxo definidas no MySQL são: IF, CASE, LOOP, LEAVE, ITERATE, REPEAT e WHILE.

- Vamos discutir as duas mais usadas, o IF, que define estruturas de decisão, e o WHILE, que define estruturas de repetição.

- Abaixo apresenta a estrutura do IF em MySQL:

```
IF condição 1 THEN
    comandos em SQL
Else
    comandos em SQL
END IF;
```

Estruturas de Procedimentos

- Exemplo:

```
DELIMITER $$
CREATE PROCEDURE nome_idade
(nome varchar(50), idade int)
BEGIN

    if((nome != '') and (idade > 0)) then
        Select nome, idade;
    else
        Select 'Nome ou idade Inválidos!';
    end if;

END$$
```

Estruturas de Procedimentos

- Exemplo:

```
Call nome_idade('Tadeu Classe', 25)
```

	nome	idade
	Tadeu Classe	25

```
Call nome_idade('', 25)
```

Nome ou idade Inválidos!
Nome ou idade Inválidos!

Estruturas de Procedimentos

- Exercícios:
  - Altere os procedimentos dos exercícios anteriores para verificar se o registros retornados para as variáveis são válidos ou não. Caso sejam válidos as inclusões podem ocorrer normalmente. Se não forem válidos, o banco de dados deve retornar uma mensagem de erro.

Estruturas de Procedimentos

- WHILE, Exemplo:

```
DELIMITER $$
CREATE PROCEDURE nome_idade
(nome varchar(50), idade int)
BEGIN
    declare cont int;

    if((nome != '') and (idade > 0)) then

        set cont = 1;

        while (cont < idade) DO

                Select nome, cont, 'Crescendo...' periodo;
                set cont = cont +1;

        End while;

        Select nome, idade, 'Cresceu!' periodo;
    else
        Select 'Nome ou idade Inválidos!';
    end if;
END$$
```

Estruturas de Procedimentos

- Exercícios:
  - Crie um procedimento que altere os salário de funcionários que tenham o título de desenvolvedor em 10% do valor;
  - Crie um procedimento para alterar o salário de todos os funcionários em 5% de acréscimo;
  - Crie um procedimento que altere todos os títulos de quem é Desenvolvedor para Engenheiro;

Estruturas de Procedimentos

- Exercícios:
  - Crie um procedimento para despedir (excluir) todos os funcionários que tenha mais de 55 anos;
  - Crie um procedimento para alterar o salário de um determinado funcionário.
  - Em todos os exercícios devem ser realizadas verificações de informações válidas;

Conclusão

- Nesta unidade, estudamos a criação e utilização dos procedimentos armazenados.

- Aprendemos a usar os comandos CREATE PROCEDURE, CALL e DROP PROCEDURE para criar, executar e apagar um procedimento.

- Aprendemos como utilizar o recurso de lista de parâmetros para definir parâmetros de entrada e saída para os procedimentos.

- Vimos também o uso dos procedimentos em conjunto com as estruturas de controle de fluxo de dados, IF ... END IF e WHILE ... END WHILE.

Obrigado e Até a Próxima Aula!

Contato:

Prof. Tadeu Classe
Bacharelado em Sistemas de Informação (BSI)
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

tadeu.classe@uniriotec.br