

Banco de Dados II



Técnicas de Programação em Banco de Dados Procedures – Parte I

Prof. Tadeu Classe
tadeu.classe@uniriotec.br

- Os procedimentos ou Stored Procedures em banco de dados são desenvolvidos para o desenvolvimento de operações úteis para a manipulação e acesso a dados.
- Em alguns casos nos encontrados em situações onde há necessidade de realizar operações complexas envolvendo os dados;
- Em outros casos, deseja-se criar operações com base em parâmetros definidos pelos usuários;

- O ideal é que as operações não sejam definidas na aplicação que fará uso do banco de dados, mas no próprio servidor;
- Isso garante a segurança dos dados, uma vez que o banco mesmo pode controlar o acesso a elas;
- As stored procedures são definidas como conjunto de comandos SQL definidos no servidor de banco de dados e acionados por eventuais chamadas geradas no banco de dados;

- Além da segurança do banco de dados, já que o usuário poderá executar os experimentos a que tiver acesso, as procedures ajudam a diminuir o tráfego de informações cliente X servidor;
- Por exemplo (banco employees): suponha que diversos usuários desejam retornar uma lista de funcionários de um departamento:

```
SELECT concat(e.first_name, ' ', e.last_name) names
FROM employees e
JOIN dept_emp de
    ON de.emp_no = e.emp_no
JOIN departments d
    ON d.dept_no = de.
dept_no
WHERE d.dept_name = 'Development'
```

- A execução deste conjunto de comandos por vários usuários simultaneamente irá ocasionar em um grande tráfego de informações por uma rede;
- Isso não ocorreria se esse conjunto de comando fossem definidos em um procedimento existente no próprio servidos de banco de dados;
- O usuário apenas irá solicitar sua execução e o retorno dos resultados executados através da chamada:

```
CALL <nome_do_procedimento>
```

Criando, Executando e Apagando Procedimentos

- A sintaxe para se criar uma stored procedure é simples, bastando usar o comando CREATE PROCEDURE e uma lista de parâmetros (caso seja necessário) a serem usados.

```
CREATE PROCEDURE <nome_do_procedimento>  
([parametros de entrada/saída])  
BEGIN  
    <comandos SQL>  
END;
```

Criando, Executando e Apagando Procedimentos

- <nome do procedimento>: nome do procedimento. É importante pois este é o nome que será executados na CALL;
- [parametros de entrada / saida] (opcional): são os parâmetros que o procedimento necessita para executar os comandos, ou retorna ao usuário;
- <comandos SQL>: comandos em SQL para a execução de uma determinada tarefa;

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento para retornar todos os funcionários de todos os departamentos:
 - 1: Criação da Consulta SQL:

```
SELECT e.first_name, e.last_name
FROM employees e
JOIN dept_emp de
    ON de.emp_no = e.emp_no
JOIN departments d
    ON d.dept_no = de.dept_no
```


Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento para retornar todos os funcionários de todos os departamentos:
 - 2: Coloque a consulta do formato de procedimento:

```
DELIMITER $$  
CREATE procedure empregado()  
BEGIN  
    SELECT e.first_name, e.last_name  
    FROM employees e  
    JOIN dept_emp de  
        ON de.emp_no = e.emp_no  
    JOIN departments d  
        ON d.dept_no = de.dept_no;  
END $$
```

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento para retornar todos os funcionários de todos os departamentos:
 - 2: Coloque a consulta do formato de procedimento:

```
DELIMITER $$  
CREATE procedure empregado()  
BEGIN  
    SELECT e.first_name, e.last_name  
    FROM employees e  
    JOIN dept_emp de  
        ON de.emp_no = e.emp_no  
    JOIN departments d  
        ON d.dept_no = de.dept_no;  
END $$
```

DELIMITER: Necessário pois o comando END necessita de uma final, no caso um delimitador.

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento para retornar todos os funcionários de todos os departamentos:
 - 3: Execução do procedimento:

```
CALL empregado();
```

first_name	last_name
Georgi	Facello
Bezael	Simmel
Parto	Bamford
Chirstian	Koblick
Kyoichi	Maliniak
Anneke	Preusig
Tzvetan	Zielinski
Saniya	Kalloufi
Sumant	Peac
.	.

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar todos os funcionários de um dos departamentos:
 - 1: Criação da Consulta SQL:

```
SELECT e.first_name, e.last_name
FROM employees e
JOIN dept_emp de
    ON de.emp_no = e.emp_no
JOIN departments d
    ON d.dept_no = de.dept_no
WHERE d.dept_name = 'Development'
```

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
- Crie um procedimento para retornar todos os funcionários de um dos departamentos:
- 2: Criação da Estrutura do Procedimento:

```
DELIMITER $$  
CREATE PROCEDURE empregados_dept  
(departamento varchar(50))  
BEGIN  
    SELECT e.first_name, e.last_name  
    FROM employees e  
    JOIN dept_emp de  
        ON de.emp_no = e.emp_no  
    JOIN departments d  
        ON d.dept_no = de.dept_no  
    WHERE d.dept_name = departamento;  
END $$
```

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar todos os funcionários de um dos departamentos:
 - 3: Chamado do procedimento:

```
CALL empregados_dept('Development');
```

first_name	last_name
Georgi	Facello
Anneke	Preusig
Saniya	Kalloufi
Patricio	Bridgland
Berni	Genin
Kazuhide	Peha
Ramzi	Erde

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 1: Criação do SQL de consulta:

```
SELECT count(*) qtde
FROM employees e
JOIN dept_emp de
    ON de.emp_no = e.emp_no
JOIN departments d
    ON d.dept_no = de.dept_no
WHERE d.dept_name = 'Development';
```

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 2: Colocar do formato de procedimento:

```
DELIMITER $$
CREATE PROCEDURE qdte_emp_dept
(departamento varchar(52), OUT quantidade bigint)
BEGIN
    SELECT count(*) qtde
    INTO quantidade
    FROM employees e
    JOIN dept_emp de
        ON de.emp_no = e.emp_no
    JOIN departments d
        ON d.dept_no = de.dept_no
    WHERE d.dept_name = departamento;
END $$
```


Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 2: Colocar do formato de procedimento:

```
DELIMITER $$
CREATE PROCEDURE qdte_emp_dept
(departamento varchar(52), OUT quantidade bigint)
BEGIN
    SELECT count(*) qtde
    INTO quantidade
    FROM employees e
    JOIN dept_emp de
        ON de.emp_no = e.emp_no
    JOIN departments d
        ON d.dept_no = de.dept_no
    WHERE d.dept_name = departamento;
END $$
```

Para saídas, é necessário que seja informada na declaração dos parâmetros como OUT

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 2: Colocar do formato de procedimento:

```
DELIMITER $$
CREATE PROCEDURE qdte_emp_dept
(departamento varchar(52), OUT quantidade bigint)
BEGIN
    SELECT count(*) qtde
    INTO quantidade
    FROM employees e
    JOIN dept_emp de
        ON de.emp_no = e.emp_no
    JOIN departments d
        ON d.dept_no = de.dept_no
    WHERE d.dept_name = departamento;
END $$
```

É preciso associar a qual parâmetro será retornado, através de INTO

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 3: Chamada do procedimento

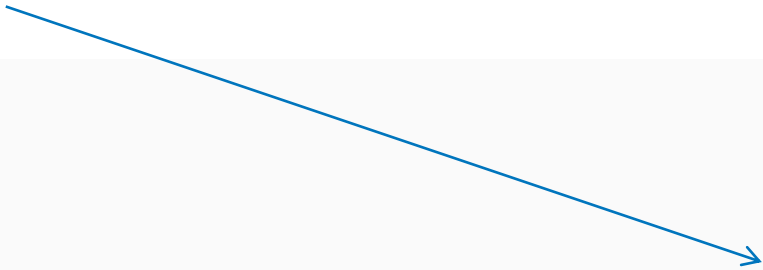
```
CALL qdte_emp_dept('Development', @qtde);  
SELECT @qtde;
```

↓
Variável para qual o
resultado do
procedimento
retornará o valor.

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Crie um procedimento para retornar o total de funcionários de um dos departamentos:
 - 3: Chamada do procedimento

```
CALL qdte_emp_dept('Development', @qtde);  
SELECT @qtde;
```



Exibição do valor
retornado pela
execução do
procedimento.

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Outro exemplo:

```
DELIMITER $$
CREATE PROCEDURE qdte_maior_emp_dept
(departamento varchar(52), OUT quantidade bigint, OUT maior date)
BEGIN
    SELECT count(*) qtde, MAX(e.birth_date) nasc
    INTO quantidade, maior
    FROM employees e
    JOIN dept_emp de
        ON de.emp_no = e.emp_no
    JOIN departments d
        ON d.dept_no = de.dept_no
    WHERE d.dept_name = departamento;
END$$
```

Criando, Executando e Apagando Procedimentos

- Exercitando: Procedimentos com Parâmetros
 - Outro exemplo:

```
CALL qdte_maior_emp_dept('Development', @qtde, @maior);  
SELECT @qtde, @maior;
```

	@qtde	@maior
	85707	1965-02-01

Criando, Executando e Apagando Procedimentos

- Apagando:
 - Os procedimento assim como as outras estruturas do banco de dados pode ser excluída com o comando DROP;

```
DROP PROCEDURE <nome do procedimento>
```

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento para retornar o salário mais alto de cada um dos departamentos.
 - Crie um procedimento que retorne o salário mais alto da empresa.
 - Crie um procedimento que retorne o nome, sobrenome, idade, salário e departamento do funcionário mais velho da empresa,

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento que retorne o nome, sobrenome, idade, salário e departamento do funcionário mais novo da empresa,
 - Crie um procedimento que mostre todos os departamentos e seus gerentes;
 - Crie um procedimento que filtre todos os funcionários de acordo com seu sexo;

Criando, Executando e Apagando Procedimentos

- Exercitando:
 - Crie um procedimento que conte a quantidade de funcionários masculinos e femininos e os retorne;
 - Crie um procedimento que exiba e retorne o título com a maior quantidade de funcionários;
 - Crie um procedimento que retorne todos os funcionários de um determinado título;
 - Crie um procedimento que filtre funcionários por título e departamento;

- Os procedimentos no banco de dados podem ser alterados de acordo com a necessidade do usuário.
- Por exemplo: temos o procedimento abaixo. E se eu quiser que ele retorne a metade da quantidade?

```
CREATE PROCEDURE qdte_emp_dept
(departamento varchar(52), OUT quantidade bigint)
BEGIN
    SELECT count(*) qtde
    INTO quantidade
    FROM employees e
    JOIN dept_emp de
      ON de.emp_no = e.emp_no
    JOIN departments d
      ON d.dept_no = de.dept_no
    WHERE d.dept_name = departamento;
END
```

- Os procedimentos no banco de dados podem ser alterados de acordo com a necessidade do usuário.
- Por exemplo: temos o procedimento abaixo. E se eu quiser que ele retorne a metade da quantidade?

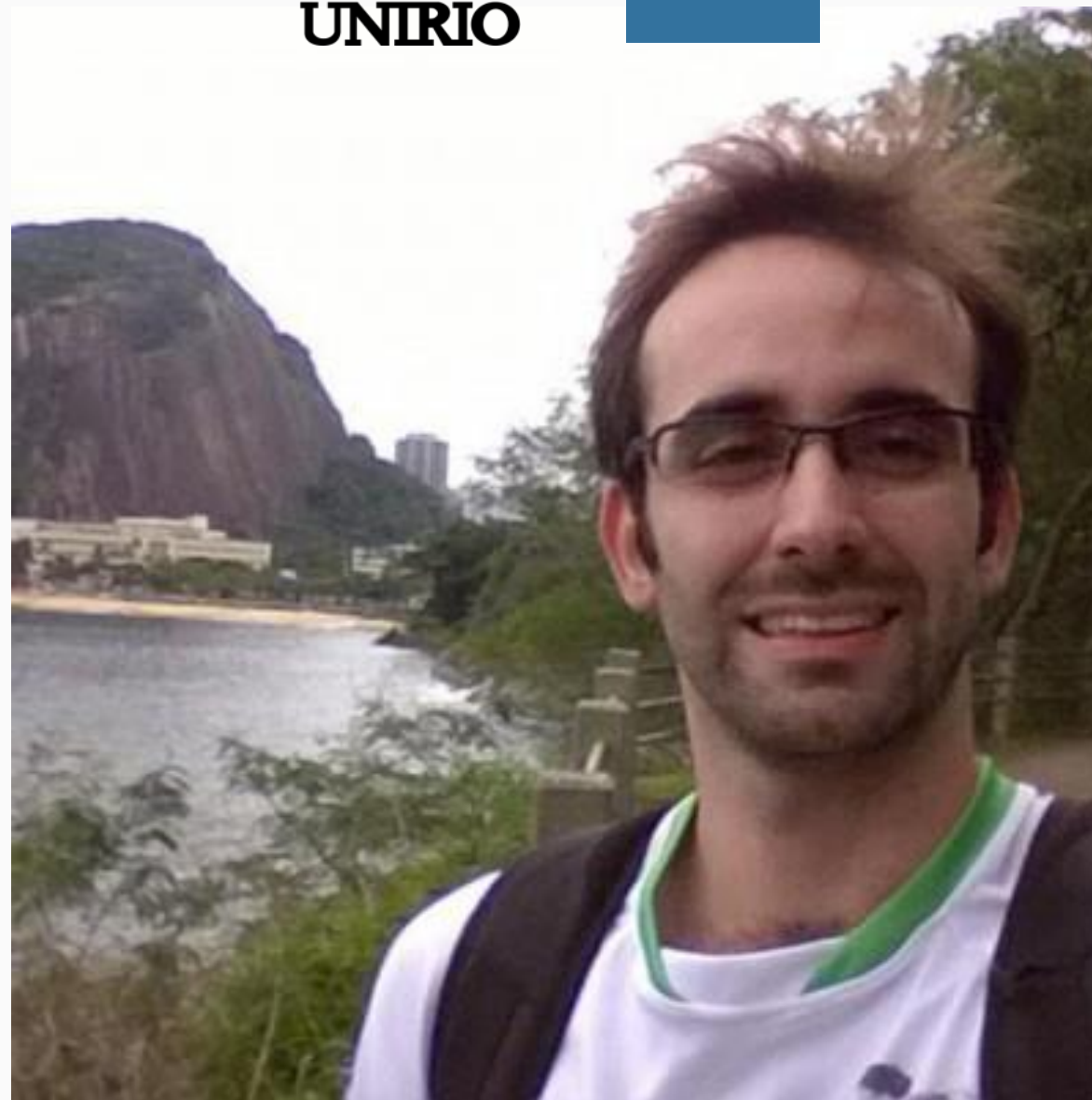
```
DROP procedure IF EXISTS `qdte_emp_dept`;  
  
DELIMITER $$  
  
CREATE PROCEDURE qdte_emp_dept  
(departamento varchar(52), OUT quantidade decimal(15.2))  
BEGIN  
    SELECT count(*) / 2 qtde  
    INTO quantidade  
    FROM employees e  
    JOIN dept_emp de  
        ON de.emp_no = e.emp_no  
    JOIN departments d  
        ON d.dept_no = de.dept_no  
    WHERE d.dept_name = departamento;  
END$$
```

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



O trabalho **Banco de Dados II - Universidade Federal do Estado do Rio de Janeiro (UNIRIO)** de Tadeu Moreira de Classe está licenciado com uma Licença [Creative Commons - Atribuição-NãoComercial-Compartilhalqual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

