

Banco de Dados II

- SQL Avançado – *Stored Procedures*

Prof. Angelo Augusto Frozza, Dr.

<http://about.me/TilFrozza>



INSTITUTO FEDERAL
Catarinense
Campus Camboriú

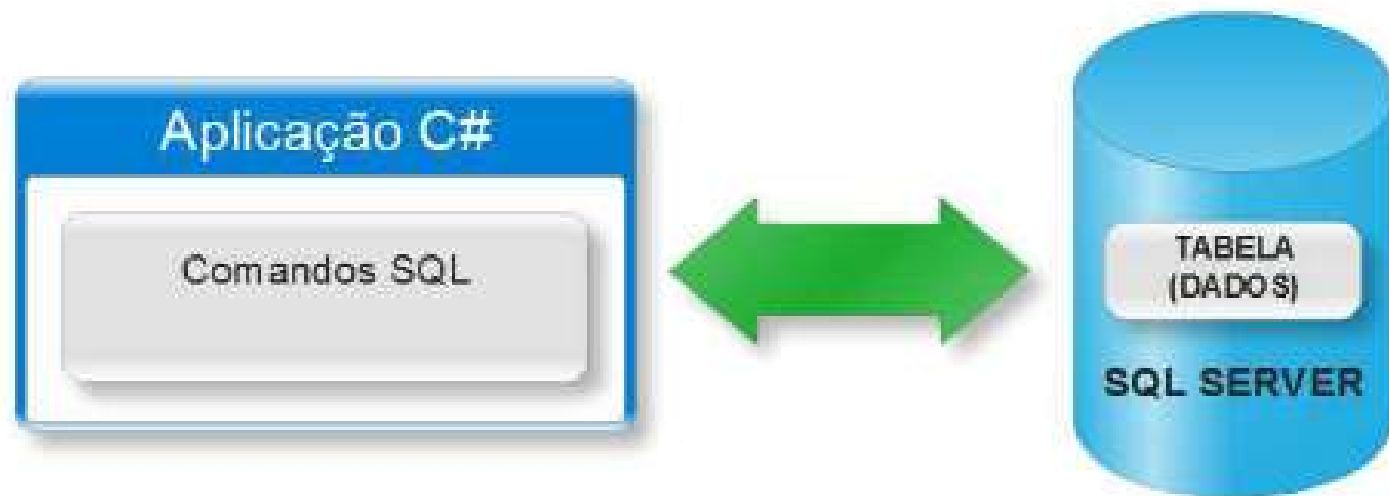


- *Stored Procedures* (Procedimentos armazenados)

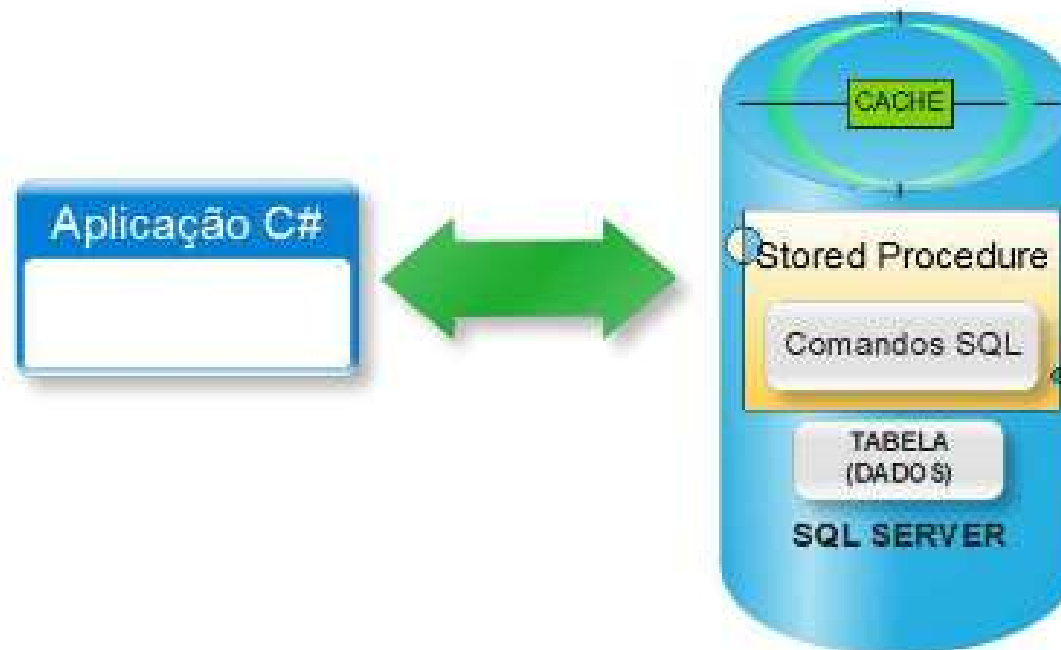
■ Procedimentos

- *Procedures*
- Procedimentos armazenados (*stored procedures*)
- Representam porções de código SQL e não SQL que ficam armazenados de forma compilada no catálogo do SGBD e são **ativados explicitamente** por aplicações, *triggers* ou outras rotinas.
- Como nas linguagens de programação, uma ***stored procedure*** realiza um processamento qualquer e **não devolve valor** (*return*) em seu final.
 - *Stored Procedures* são processamentos de tarefas da aplicação que residem no SGBD ao invés de estar no código da aplicação (cliente).

Modelo de Acesso ao Banco de Dados sem utilização de Stored Procedures



Modelo de Acesso ao Banco de Dados utilizando Stored Procedures



Stored Procedures

6

► Vantagens:

► Desempenho

► Ex.: Dada a consulta:

```
SELECT codigop, nome, COUNT(*)  
FROM Projeto p, Allocacao a  
WHERE p.codproj = a.codigop  
GROUP BY p.codproj, p.nome
```

- Se vários usuários realizarem esta consulta o tráfego de rede será alto.
- Com uma *stored procedure* para executar esta consulta, os usuários necessitarão apenas de um comando:
EXEC nomeProcedimento;
- A consulta anterior é compilada a cada chamada, enquanto o procedimento contendo a consulta é compilado uma única vez.

Stored Procedures

► Vantagens:

► Facilita o gerenciamento do BD

- A consulta é escrita em um único lugar, portanto a manutenção desta torna-se mais eficaz e segura.
- Aumenta o reaproveitamento de código e melhora a modularidade em contextos em que um banco de dados é utilizado por várias aplicações.
- Facilita a manipulação de tipos de dados complexos usados pelos procedimentos.

► Segurança

- Podemos usar *stored procedures* para limitar o acesso de alguns usuários ao BD.
- Desta forma, a maneira em que o BD pode ser modificado é estritamente definida.

Stored Procedures

► Implementação:

- **Stored Procedures** podem ser implementadas de vários modos:
 - Linguagens não-procedurais
 - **Procedurais** – seguem o padrão SQL/PSM (ISO *standard*)
 - Linguagens externas – geralmente C++

Stored Procedures

➤ Implementação:

➤ SQL/PSM - *Persistent Stored Modules*

- Cada SGBD oferece sua própria linguagem (*Oracle PL/SQL*, *Microsoft Transact/SQL*, ***PostgreSQL PL/pgSQL*** etc.)
- Em PSM, define-se módulos que são coleções de definições de funções ou procedimentos, declarações de tabelas temporárias, entre outros.

Stored Procedures no PostgreSQL

- Características das *Stored Procedures* no *PostgreSQL*
 - SPs não retornam valores
 - SPs utilizam *plpgsql*

Stored Procedures no PostgreSQL

➤ Sintaxe:

CREATE [OR REPLACE] FUNCTION

name ([[**argmode**] [**argname**] **argtype** [{ **DEFAULT** | = }
 default_expr] [, ...]])

[**RETURNS** **rettype**

| **RETURNS TABLE** (**column_name** **column_type** [, ...])]

{ **LANGUAGE** **lang_name**

| **WINDOW**

| **IMMUTABLE** | **STABLE** | **VOLATILE** | [**NOT**] **LEAKPROOF**

| **CALLED ON NULL INPUT** | **RETURNS NULL ON NULL INPUT** | **STRICT**

| [**EXTERNAL**] **SECURITY INVOKER** | [**EXTERNAL**] **SECURITY DEFINER**

| **COST** **execution_cost**

| **ROWS** **result_rows**

| **SET** **configuration_parameter** { **TO** **value** | = **value** | **FROM CURRENT** }

| **AS** 'definition'

| **AS** 'obj_file', 'link_symbol'

} ...

[**WITH** (**attribute** [, ...])]

Stored Procedures no PostgreSQL

➡ Sintaxe (mais simples):

```
CREATE OR REPLACE FUNCTION
```

```
    funcName(varName1 varType1,...)
```

```
RETURNS returnVarType AS $$
```

```
DECLARE                                (optional)
```

```
    /* All Variables Declared Here*/
```

```
BEGIN                                (mandatory)
```

```
    /* Executable statements  
    (what the block DOES!)*/*
```

```
EXCEPTION                            (optional)
```

```
    /* Exception handling*/
```

```
END;                                (mandatory)
```

```
$$ LANGUAGE 'plpgsql';
```

Stored Procedures no PostgreSQL

➤ Exemplo (mais direto):

```
CREATE OR REPLACE FUNCTION
```

```
  InsererFuncionario (a_codigo INTEGER,  
                      a_nome VARCHAR(100), a_email VARCHAR(150),  
                      a_telefone VARCHAR(15), a_cidade VARCHAR(50),  
                      a_estado VARCHAR(2))
```

```
RETURNS void AS $$
```

```
BEGIN
```

```
  INSERT INTO tb_funcionários (codigo, nome, email,  
                               telefone, cidade, estado)  
  VALUES (a_codigo, a_nome, a_email, a_telefone,  
          a_cidade, a_estado);
```

```
END;
```

```
$$ LANGUAGE 'plpgsql';
```

Stored Procedures

Exercícios

- Crie as seguintes *stored procedures* na base de testes da *Clínica Médica*:
 - Efetuar a inclusão de registros nas tabelas `Ambulatorio`, `Doencas`, `Funcionarios`, `Pacientes`;
 - Efetuar a alteração da data e hora de uma `Consulta`;
 - Associar uma doença à um `Paciente`.



➡ Prof. Angelo Augusto Frozza, Dr.



angelo.frozza@ifc.edu.br

<http://www.ifc-camboriu.edu.br/~frozza>



@TilFrozza

<http://www.twitter.com/TilFrozza>

<http://about.me/TilFrozza>