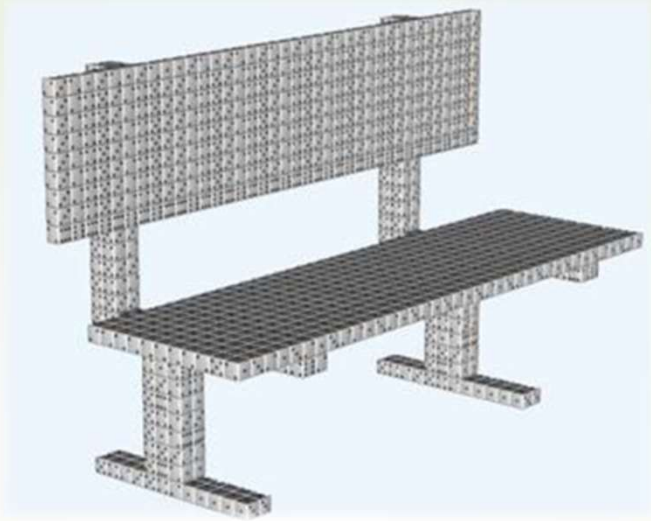

Banco de Dados II
- SQL Avançado - *Functions*

Prof. Angelo Augusto Frozza, Dr.

http://about.me/TilFrozza

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Roteiro

- *Functions* (Funções)

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- Possuem uma estrutura muito semelhante aos *Stored Procedures*;
  - Diferenças:
    - ***Functions*** sempre retornam um valor, enquanto *Stored Procedures* retornam valores através de parâmetros **OUT** explicitamente declarados;
    - Podem ser usadas de forma direta dentro de comandos SQL, enquanto que *Stored Procedures* devem ser usados através de um comando de chamada.

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- O objetivo das ***Functions*** não é realizar transações de negócio completas, como os ***Stored Procedures***, mas sim, realizar pequenas operações auxiliares.
  - Por exemplo:
    - Formatações de textos e variáveis;
    - Operações repetitivas e rotineiras que possam ser compartilhadas por diversos *Procedures*;
  - **No *PostgreSQL*, *Functions* funcionam da mesma forma como *Stored Procedures*, com o diferencial que, conceitualmente, retornam valores.**

INSTITUTO FEDERAL Catarinense Campus Camboriú

Functions

- ***PostgreSQL* – funções pré-definidas**
  - O *PostgreSQL* possui um conjunto completo de funções pré-definidas: matemáticas, para *string*, binários, *bits*, formatação e conversão de tipos de dados, geométricas, XML, *arrays* etc.

  - Exemplos:
    - *Abs(x)* - retorna o valor absoluto
    - *Ln(x)* - logaritmo natural
    - *Mod(x, y)* - resto de *x / y*
    - *Pi()* - valor de *PI*
    - *Power(a,b)* - *a* elevado a *b*
    - *Random()* - valor randômico entre 0.0 e 1.0

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- ***PostgreSQL* – funções pré-definidas**
  - Exemplos:
    - *Avg(expression)* - média aritmética
    - *Count(expression)* - número de registros
    - *Max(expression)* - valor máximo
    - *Sum(expression)* - soma de valores
    - *Bit_and(expression)* - comparação *AND bit* a *bit*
    - *Xmlelement(...)* - para criar elementos XML
    - *Lower(string)* - converte *string* para caixa baixa
    - *Upper(string)* - converte *string* para caixa alta
    - *Concat(...)* - concatena *strings*
    - *Length(String)* - número de caracteres de *string*

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

PostgreSQL – funções de usuário

```
CREATE [ OR REPLACE ] FUNCTION
    name( [ [ argname] argtype] )
    [ RETURNS tipo | [TABLE (cols)] ] AS $$
    [DECLARE var tipo ;]
    BEGIN
        operações
    END;
  $$ LANGUAGE 'plpgsql' ;
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

Exemplos

```
CREATE OR REPLACE FUNCTION
    soma ( a INTEGER, b INTEGER )
RETURNS INTEGER AS $$
DECLARE result INTEGER ;
BEGIN
 result := a + b ;
    RETURN result ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

Exemplos – usando condicional

```
CREATE OR REPLACE FUNCTION
    numeroPar ( a INTEGER )
RETURNS BOOLEAN AS $$
DECLARE temp INTEGER ;
BEGIN
 temp := A % 2 ;
   IF temp = 0
      THEN RETURN true ;
      ELSE RETURN false ;
   END IF ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

Exemplos – laço FOR

```
CREATE OR REPLACE FUNCTION
    fatorial ( a NUMERIC )
RETURNS NUMERIC AS $$
DECLARE temp NUMERIC ;
        result NUMERIC ;
BEGIN
 result := 1 ;
   FOR temp IN 1 .. a LOOP
      result := result * temp ;
   END LOOP ;
   RETURN result ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- Exemplos – laço WHILE

```
CREATE OR REPLACE FUNCTION fatorial ( a NUMERIC )
RETURNS NUMERIC AS $$
DECLARE temp NUMERIC ;
        result NUMERIC ;
BEGIN
 result := 1 ;
   temp := 1 ;
   WHILE temp <= a LOOP
      result := result * temp ;
      temp := temp + 1 ;
   END LOOP ;
   RETURN result ;
END;
$$ LANGUAGE 'plpgsql';
```

Functions

INSTITUTO FEDERAL Catarinense Campus Camboriú

- Exemplos – SQL dinâmico

```
CREATE OR REPLACE FUNCTION
   recuperaFuncionario ( id INTEGER )
RETURNS funcionario AS $$
DECLARE registro RECORD ;
BEGIN
   EXECUTE 'SELECT * FROM funcionarios
                  WHERE id = ' || id INTO registro ;
   RETURN registro ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- Exemplos – cursor

```
CREATE OR REPLACE FUNCTION totalSalarios ( )
RETURNS NUMERIC AS $$
DECLARE registro RECORD ;
        result NUMERIC ;
BEGIN
    result := 0 ;
    FOR registro IN SELECT * FROM funcionario LOOP
       result := result + registro.salario ;
    END LOOP ;
    RETURN result ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Functions

- Exemplos – Retornando apenas UM ÚNICO VALOR na consulta

```
CREATE OR REPLACE FUNCTION
      buscaEmpregadoPorNome ( myname VARCHAR )
RETURNS funcionario AS $$
DECLARE registro RECORD ;
BEGIN
   SELECT * INTO STRICT registro FROM emp
       WHERE empname = myname ;
    RETURN registro ;
    EXCEPTION
       WHEN NO_DATA_FOUND THEN
          RAISE EXCEPTION 'funcionario % não encontrado', myname ;
       WHEN TOO_MANY_ROWS THEN
          RAISE EXCEPTION 'funcionario % não único', myname ;
END;
$$ LANGUAGE 'plpgsql';
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Exercícios

- Uma prática utilizada durante o desenvolvimento de aplicações que interagem com BDs é a de definir procedimentos ou funções responsáveis pela inclusão, alteração e exclusão de registros.
- Para as tabelas de **`Ambulatorios`**, **`Funcionarios`**, **`Doencas`** e **`Pacientes`**, crie funções que atendam a essas operações, respeitando as seguintes regras:
  a) no caso de **inclusões**, a função deve retornar a chave primária do novo registro como resultado;
  b) no caso de **alterações**, a chave primária cujo registro deve ser modificado deve ser passada como parâmetro (juntamente com os dados a serem modificados no registro). O retorno dessa função deve ser nulo;
  c) no caso de **exclusões**, a chave primária cujo registro deve ser removido deve ser passada como parâmetro. O retorno dessa função deve ser nulo.

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Exercícios

- Crie uma função que realiza o agendamento de uma Consulta.
  - A função deve receber o nome e o CPF do paciente, a data e a hora da consulta e o CPF do médico.
  - Caso o paciente não esteja cadastrado, a função deve inserir o paciente.

Contato

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- 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