

# BASE DE DADOS



Oracle  
PL/SQL – Funções  
e Procedimentos

Teórico-Práticas  
Ano Lectivo 2016/2017  
Rosa Reis



# INTRODUÇÃO

- ★ Funções e procedimentos são blocos PL/SQL armazenados de forma compilada, e executados no servidor
- ★ Características de um subprograma
  - nome
    - ✧ máximo de 30 caracteres
  - Modo dos parâmetros
    - ✧ **IN** (*de entrada*; passagem por valor)
      - ✓ Passa um valor que não pode ser alterado pelo subprograma
    - ✧ **OUT** (*de saída*; passagem por referência)
      - ✓ Valor de retorno. Deve ser inicializado no subprograma
    - ✧ **IN OUT** (*de entrada/saída*; passagem por referência)
      - ✓ Passa um valor e retorna o valor atualizado pelo subprograma
  - não possui o comando DECLARE



# PROCEDURES

- ✦ Subprogramas que têm por objetivo executar uma ação específica
- ✦ Restrições
  - não são utilizados em atribuições
  - não podem ser usados como argumentos de um comando SELECT

```
CREATE OR REPLACE PROCEDURE nome_proc  
  (parâmetro1 modo tipo_de_dados,  
   parâmetro2 modo tipo_de_dados,  
   ...  
   parâmetro3 modo tipo_de_dados)  
IS ou AS  
  área de declaração  
BEGIN  
  área de comandos  
END nome_proc;
```



# PROCEDURES

★ Exemplo:

```
create or replace PROCEDURE award_bonus (emp_id NUMBER) IS
    bonus      REAL;

BEGIN
    SELECT comm*0.15 INTO bonus
      FROM emp WHERE empno = emp_id;

    IF bonus IS NOT NULL THEN
        UPDATE payroll SET pay = pay + bonus
          WHERE empno = emp_id;
    END IF;

END award_bonus;
```

**Invocado como um comando PL/SQL:**

```
BEGIN
    ...
    award_bonus(33); -> chamada do
    procedimento
    ...
END;
```



# FUNCTIONS

- ✳ Subprogramas que têm por objetivo retornar um resultado
- ✳ Funcionalidades
  - podem ser usadas em atribuições
  - podem ser usadas como argumentos de um comando SELECT
  - Podem ser recursivas

```
CREATE OR REPLACE FUNCTION nome_func  
  (parâmetro1 modo tipo_de_dados,  
   parâmetro2 modo tipo_de_dados,  
   ...  
   parâmetro3 modo tipo_de_dados)  
RETURN tipo_de_dados  
IS ou AS  
  área de declaração  
BEGIN  
  área de comandos  
END nome_func;
```



# FUNCTIONS

## ★ Exemplo:

```
CREATE OR REPLACE FUNCTION soma_numeros (telefone IN VARCHAR2)
RETURN NUMBER IS
S number(4) := 0;
Soma number(4):= 0;
BEGIN
    FOR X IN 1..15
    LOOP
        IF Substr(telefone, x, 1) IN ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9')
        THEN s := to_number(Substr(telefone, x, 1));
            soma := soma + s;
        END IF;
    END LOOP;
    RETURN soma;
END soma_numeros;
```





# FUNCTIONS - CHAMADA

```
set serveroutput on;
```

```
DECLARE
```

```
telefone varchar(15):= '(016) 3373-3373';
```

```
soma NUMBER(4);
```

```
BEGIN
```

```
    dbms_output.put_line ( 'O telefone de entrada é: ' || telefone );
```

```
    soma := soma_numeros (telefone);
```

```
    dbms_output.put_line ( 'A soma dos dígitos do telefone é: ' || soma );
```

```
END;
```



# EXERCÍCIOS

Elabore uma função em PL/SQL para calcular o factorial de um número.

```
create or replace function factorial ( n in number) return number
```

```
is
```

```
    fact number(3);
```

```
begin
```

```
    fact:=1;
```

```
    for i in 1..n loop
```

```
        fact:=fact*i;
```

```
    end loop;
```

```
    return(fact);
```

```
end;
```

## Chamada da Função

**Declare**

```
    resfact number(3);
```

**Begin**

```
    resFact := factorial (3);
```

```
    dbms_output.put_line(' O factorial de 3 ' || resfact);
```

**End;**

**set serveroutput on**





# EXERCÍCIOS

Elabore **PROCEDIMENTO** em PL/SQL para calcular o factorial de um número

**create or replace procedure factorial** ( n in number) **is**

fact number(3);

**begin**

fact:=1;

for i in 1..n loop

fact:=fact\*i;

end loop;

dbms\_output.put\_line(' O factorial de 3 ' ||

fact);

**end;**

## **Chamada do Procedimento**

**Declare**

resfact number(3);

**Begin**

factorial (3);

**End;**



# EXERCÍCIOS

Elabore PROCEDIMENTO em PL/SQL para calcular o factorial de um numero- **VERSÃO 2**

create or replace procedure factorial1 ( n in number, **fact in out number**) is

**begin**

**for i in 1..n loop**

fact:=fact\*i;

**end loop;**

**end;**

## Chamada do Procedimento

declare

fact number(3);

begin

fact:=1;

**factorial1(6, fact);**

dbms\_output.put\_line(fact);

end;



# Funções do Oracle

## ★ Funções DATE

- Armazena a data num formato numérico interno : century, year, month, day;
- Por omissão o formato é : DD-MON- YY;
- Sysdate é uma função que retorna a data e hora;

Function	Result
<b>MONTHS_BETWEEN</b>	<b>Number of months between two dates</b>
<b>ADD_MONTHS</b>	<b>Add calendar months to date</b>
<b>NEXT_DAY</b>	<b>Next day of the date specified</b>
<b>LAST_DAY</b>	<b>Last day of the month</b>
<b>ROUND</b>	<b>Round date</b>
<b>TRUNC</b>	<b>Truncate date</b>

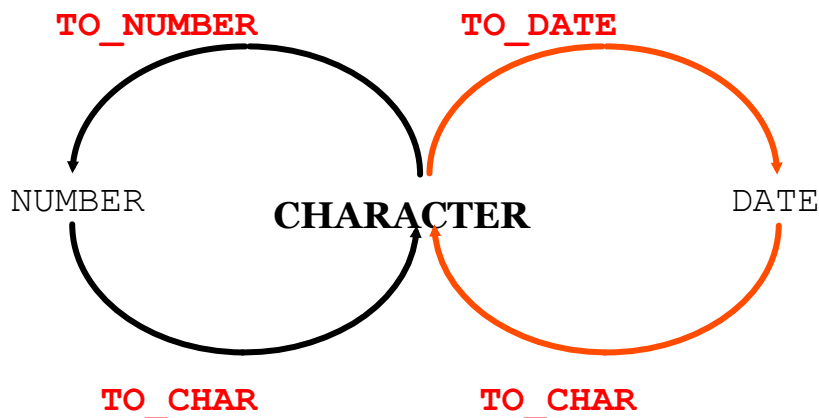


# Funções do Oracle

## ★ Funções de conversão

Conversão implícita de tipo de dados →

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE



→ Conversão explícita de tipo de dados



# Funções do Oracle

## ★ **TO\_CHAR** (date, 'format\_model')

- Converte uma data para uma string usando o formato especificado.

```
SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD') FROM dual;
```

**Format\_model** →

Element	Result
YYYY	Full year in numbers
YEAR	Year spelled out (in English)
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month



# Funções do Oracle

## ★ **TO\_DATE** (char [, 'format\_model'] )

- Converte uma string para um formato de data.

```
SELECT TO_DATE('2015/05/15 8:30:25', 'YYYY/MM/DD HH:MI:SS')  
FROM dual;
```

## ★ **TO\_EXTRACT**

- Permite extrair um valor a partir de uma data ou intervalo.

```
EXTRACT (  
  { YEAR | MONTH | DAY | HOUR | MINUTE | SECOND }  
  | { TIMEZONE_HOUR | TIMEZONE_MINUTE }  
  | { TIMEZONE_REGION | TIMEZONE_ABBR }  
FROM { date_value | interval_value } )
```





# Funções do Oracle

## ★ Exemplos:

```
SELECT EXTRACT(YEAR FROM DATE '2003-08-22') FROM dual;  
Resultado: 2003
```

```
SELECT EXTRACT(DAY FROM DATE '2011-01-12') FROM dual;  
Resultado: 12
```

```
SELECT EXTRACT(HOUR FROM SYSDATE) FROM dual;  
-- ERROR at line 1:  
-- ORA-30076: invalid extract field for extract source
```

```
SELECT EXTRACT(HOUR FROM TIMESTAMP '2012-01-12 10:11:00') FROM dual;  
Resultado: 10
```