

Verificar a Existência de Sequência

/*

AUTOR : JOSE FABIO N. ALMEIDA

DATA : 29/03/04

RESPONSABILIDADE : VERIFICA SE EXISTE UM SEQUENCE NO BANCO

E-MAIL: fabiorecife@yahoo.com.br

RETORNA: TRUE se existe

FALSE se não existe

*/

CREATE OR REPLACE FUNCTION EXIST_SEQUENCE (VARCHAR) RETURNS BOOLEAN
AS '

DECLARE

VCOUNT INT4;

SEQUENCE_NAME ALIAS FOR \$1;

BEGIN

SELECT INTO VCOUNT COUNT (*) FROM PG_CLASS WHERE RELNAME = cast
(SEQUENCE_NAME as name)

AND (RELKIND = "S");

RETURN VCOUNT > 0;

END;

' LANGUAGE 'plpgsql';

Usando:

SELECT exist_sequence('nometabela');

Checar Existência de Tabela

/*

AUTOR : JOSE FABIO N. ALMEIDA

DATA : 29/03/04

RESPONSABILIDADE : VERIFICA SE EXISTE UMA TABELA NO BANCO

E-MAIL: fabiorecife@yahoo.com.br

RETORNA: TRUE se existe

FALSE se não existe

*/

CREATE OR REPLACE FUNCTION EXIST_TABLE (VARCHAR) RETURNS BOOLEAN AS '
DECLARE

VCOUNT INT4;

TABLE_NAME ALIAS FOR \$1;

BEGIN

SELECT INTO VCOUNT COUNT (*) FROM PG_CLASS WHERE RELNAME = cast
(TABLE_NAME as name)

AND (RELKIND = "r" OR RELKIND = "t");

RETURN VCOUNT > 0;

END;

' LANGUAGE 'plpgsql';

Usando:

```
SELECT exist_table('test');
```

Exibir Números por Extenso

```
CREATE or replace FUNCTION FU_EXTENSO_REAL(num numeric(20,2)) returns text as $$
-- num -> numero a ser convertido em extenso
begin
    return fu_extenso(num, 'Real', 'Reais') ;
end ;
$$ LANGUAGE plpgsql
IMMUTABLE
RETURNS NULL ON NULL INPUT ;
```

```
CREATE or replace FUNCTION FU_EXTENSO(num numeric(20,2) , moeda text , moedas
text) returns text as $$
-- num -> numero a ser convertido em extenso
-- moeda -> nome da moeda no singular
-- moedas -> nome da moeda no plural
declare
w_int char(21) ;
x integer ;
v integer ;
w_ret text ;
w_ext text ;
w_apoio text ;
m_cen text[] :=
array['Quatrilhão', 'Quatrilhões', 'Trilhão', 'Trilhões', 'Bilhão', 'Bilhões', 'Milhão',
'Milhões', 'Mil', 'Mil'] ;
begin
    w_ret := '' ;
    w_int := to_char(num * 100 , 'fm000000000000000000 00') ;
    for x in 1..5 loop
        v := cast(substr(w_int, (x-1)*3 + 1, 3) as integer) ;
        if v > 0 then
            if v > 1 then
                w_ext := m_cen[(x-1)*2+2] ;
            else
                w_ext := m_cen[(x-1)*2+1] ;
            end if ;
            w_ret := w_ret || fu_extenso_blk(substr(w_int, (x-1)*3 + 1, 3)) || ' ' ||
w_ext || ',' ;
        end if ;
    end loop ;
    v := cast(substr(w_int, 16, 3) as integer) ;
    if v > 0 then
        if v > 1 then
            w_ext := moedas ;
        else
            if w_ret = '' then
                w_ext := moeda ;
            else
                w_ext := moedas ;
            end if ;
        end if ;
    end if ;
    w_apoio := fu_extenso_blk(substr(w_int, 16, 3)) || ' ' || w_ext ;
    if w_ret = '' then
        w_ret := w_apoio ;
    else
```

```

        if v > 100 then
            if w_ret = '' then
                w_ret := w_apoio ;
            else
                w_ret := w_ret || w_apoio ;
            end if ;
        else
            w_ret := btrim(w_ret, ', ' ) || ' e ' || w_apoio ;
        end if ;
    end if ;
else
    if w_ret <> '' then
        if substr(w_int,13,6) = '000000' then
            w_ret := btrim(w_ret, ', ' ) || ' de ' || moedas ;
        else
            w_ret := btrim(w_ret, ', ' ) || ' ' || moedas ;
        end if ;
    end if ;
end if ;
v := cast(substr(w_int,20,2) as integer) ;
if v > 0 then
    if v > 1 then
        w_ext := 'Centavos' ;
    else
        w_ext := 'Centavo' ;
    end if ;
    w_apoio := fu_extenso_blk('0' || substr(w_int,20,2)) || ' ' || w_ext ;
    if w_ret = '' then
        w_ret := w_apoio ;
    else
        w_ret := w_ret || ' e ' || w_apoio ;
    end if ;
end if ;
return w_ret ;
end ;
$$ LANGUAGE plpgsql
IMMUTABLE
RETURNS NULL ON NULL INPUT ;

```

```

CREATE or replace FUNCTION FU_EXTENSO_BLK(num char(3)) returns text as $$
declare
w_cen integer ;
w_dez integer ;
w_dez2 integer ;
w_uni integer ;
w_tcen text ;
w_tdez text ;
w_tuni text ;
w_ext text ;
m_cen text[] :=
array['','Cento','Duzentos','Trezentos','Quatrocentos','Quinhentos','Seiscentos',
'Setecentos','Oitocentos','Novecentos'];
m_dez text[] :=
array['','Dez','Vinte','Trinta','Quarenta','Cinquenta','Sessenta','Setenta','Oit
enta','Noventa'] ;
m_uni text[] :=
array['','Um','Dois','Três','Quatro','Cinco','Seis','Sete','Oito','Nove','Dez','
Onze','Doze','Treze','Quatorze','Quinze','Dezesseis','Dezessete','Dezoito','Deze
nove'] ;
begin
    w_cen := cast(substr(num,1,1) as integer) ;
    w_dez := cast(substr(num,2,1) as integer) ;

```

```

w_dez2 := cast(substr(num,2,2) as integer) ;
w_uni := cast(substr(num,3,1) as integer) ;
if w_cen = 1 and w_dez2 = 0 then
    w_tcen := 'Cem' ;
    w_tdez := '' ;
    w_tuni := '' ;
else
    if w_dez2 < 20 then
        w_tcen := m_cen[w_cen + 1] ;
        w_tdez := m_uni[w_dez2 + 1] ;
        w_tuni := '' ;
    else
        w_tcen := m_cen[w_cen + 1] ;
        w_tdez := m_dez[w_dez + 1] ;
        w_tuni := m_uni[w_uni + 1] ;
    end if ;
end if ;
w_ext := w_tcen ;
if w_tdez <> '' then
    if w_ext = '' then
        w_ext := w_tdez ;
    else
        w_ext := w_ext || ' e ' || w_tdez ;
    end if ;
end if ;
if w_tuni <> '' then
    if w_ext = '' then
        w_ext := w_tuni ;
    else
        w_ext := w_ext || ' e ' || w_tuni ;
    end if ;
end if ;
return w_ext ;
end ;
$$ LANGUAGE plpgsql
IMMUTABLE
RETURNS NULL ON NULL INPUT ;

```

Usando: select fu_extenso_real(123456.67);
 Autor: Argemiro Braga Guara Junior

Retorna o Próximo Mês

```

/*
AUTOR : JOSE FABIO N. ALMEIDA
DATA : 29/03/04
RESPONSABILIDADE : RETORNA O PROXIMO MÊS
E-MAIL: fabiorecife@yahoo.com.br
RETORNA: TRUE se existe
FALSE se não existe
*/

```

```

CREATE OR REPLACE FUNCTION NEXTMONTH (date) RETURNS date AS '
BEGIN
    RETURN $1 + INTERVAL "1 MONTH";
END;
' LANGUAGE 'plpgsql' ;

```

/*

AUTOR : JOSE FABIO N. ALMEIDA

DATA : 29/03/04

RESPONSABILIDADE : RETORNA O MÊS ANTERIOR

E-MAIL: fabiorecife@yahoo.com.br

RETORNA: TRUE se existe

FALSE se não existe

*/

CREATE OR REPLACE FUNCTION PRIORMONTH (date) RETURNS date AS '

BEGIN

RETURN \$1 + INTERVAL "-1 MONTH";

END;

' LANGUAGE 'plpgsql' ;