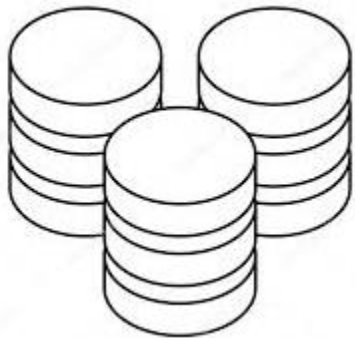


Banco de Dados II

A Linguagem PL/pgSQL Funções e Exceções



Profa. Damires Souza
damires@ifpb.edu.br



Variáveis integradas

- **FOUND**

- **SELECT INTO**: define **FOUND** como **true** se retornar uma linha e **false** se não retornar linhas.
- **UPDATE, INSERT e DELETE**: define **FOUND** como **true** se pelo menos uma linha for afetada e como **false** se nenhuma linha for afetada.

- **SQLSTATE**

- Contém o **código de erro** que corresponde à exceção gerada

- **SQLERRM**

- Possui a **mensagem de erro** associada à exceção

- **GET DIAGNOSTICS integer_var := ROW_COUNT;**

- Mostra o **número de linhas** processadas pelo último comando SQL enviado.

Variável FOUND [NOT FOUND]

Pode ser verificada imediatamente após a instrução **SELECT INTO** para determinar se a atribuição foi bem-sucedida (se retornou uma linha)

Create or replace function verificaArt(codigo integer)

Returns varchar

As \$\$

Declare nome varchar(25);

Begin

select nomeart into nome from artista where codart = codigo;

IF NOT FOUND THEN


Return 'Nenhum artista com esse código foi encontrado.';

END IF;

return nome;


End; \$\$ LANGUAGE 'plpgsql';

Select verificaart(1);



	verificaart	
	character varying	
1	Cameron Diaz	

select verificaart(111);



	verificaart	
	character varying	
1	Nenhum artista com esse código foi encontrado.	

FOUND [NOT FOUND] x Exceção

- Esta condição **não é um erro** mas pode-se estar interessado em tratá-la de **forma especial** (como uma **exceção**)

```
create or replace function verificaart2(integer)
```

```
returns varchar
```

```
As $$
```

```
Declare r record;
```

```
begin
```

```
select into r * from artista where codart = $1;
```

```
if not found then RAISE EXCEPTION 'Artista não existente--> %', $1
```

```
    USING HINT = 'Por favor, verifique o código do artista';
```

```
end if;
```

```
return r.nomeart;
```

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

```
select verificaart2(1);  
select verificaart2(111);
```

Data Output Explain Messages Notifications

ERROR: Artista não existente--> 111

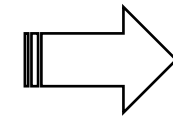
HINT: Por favor, verifique o código do artista

CONTEXT: função PL/pgSQL verificaart2(integer) linha 5 em RAISE

SQL state: P0001

Erros/exceções

Tipo de Erro	Comunicação	Modo como é tratado
Em tempo de compilação	Compilador PL/SQL	O compilador comunica os erros, e o desenvolvedor os corrige.
Runtime	Máquina de runtime PLpgSQL	São abertas exceções que são detectadas por rotinas de tratamento de exceções



ERRO



- Uma **exceção** representa uma **interrupção anormal** no processamento de uma função.

Exceções

- Exceção gerada pelo postgres
 - Quando acontece um **erro** no PostgreSQL, a exception associada é sinalizada automaticamente
- Exceção gerada intencionalmente pelo desenvolvedor
 - Uma **exception** é explicitamente sinalizada através do comando **RAISE** dentro do bloco PL/pgSQL.



Exceções no PgPLSQL

```
[ <<label>> ]  
  [ DECLARE  
    declarações ]  
  BEGIN  
    afirmações  
  EXCEPTION  
    WHEN condição [ OR condição ... ] THEN  
      Faça algo  
    [ WHEN condição [ OR condição ... ] THEN  
      Faça algo  
    ... ]  
  END;
```



Exceções pré-definidas

**** Exceção gerada pelo postgres**

Class P0 — PL/pgSQL Error	
P0000	plpgsql_error
P0001	raise_exception
P0002	no_data_found
P0003	too_many_rows
P0004	assert_failure
Class XX — Internal Error	
XX000	internal_error
XX001	data_corrupted
XX002	index_corrupted



<https://www.postgresql.org/docs/current/errcodes-appendix.html>

Exemplo- Exceção pré-definida

```
CREATE OR REPLACE FUNCTION adicione(a integer, b
integer)
RETURNS integer AS $$
BEGIN
    RETURN a + b;
    EXCEPTION
        WHEN numeric_value_out_of_range THEN
            -- fazer algo
            RETURN -1;
        WHEN OTHERS THEN
            -- fazer algo
            RETURN -1;
END;
$$ LANGUAGE plpgsql;
```

```
select adicione(1,2);
```

Exemplo – Exceção pré-definida

```
CREATE OR REPLACE FUNCTION testains(cod integer,nome varchar(25))
RETURNS integer AS $$
BEGIN
    Insert into estudio(codest,nomeest) values (cod,nome);
    Return 1;
    EXCEPTION
    WHEN unique_violation THEN
        raise notice 'Já existe esse registro';
        return -1;
    WHEN OTHERS THEN
        -- fazer algo
        RETURN -1;
END;
$$ LANGUAGE plpgsql;
```

```
select * from estudio;
select testains(4,'Teste');
select testains(5,'Teste');
```

Data Output		Explain	Messages	History
	testains			
	integer			
1	-1			

Data Output	Explain	Messages	History
NOTICE: Já existe esse registro			
Total query runtime: 13 msec			
1 row retrieved.			

Exemplo

Banco Pedidos

```
CREATE OR REPLACE FUNCTION retornaInfoProd(v_cod in produto.codprod%type)
```

```
RETURNS varchar AS $$
```

```
Declare
```

```
    V_nome produto.descricao%type;
```

```
    V_preco produto.valor%type;
```

```
    Msg varchar(30);
```

```
Begin
```

```
    Select descricao, valor into strict v_nome, v_preco From produto
```

```
    Where codprod = v_cod;
```

```
    select v_nome || '--' || v_preco into msg;
```

```
    return msg;
```

```
Exception
```

```
    When No_data_found then
```

```
        msg = 'Nenhum produto encontrado';
```

```
        Return msg;
```

```
    When others then
```

```
        Return 'Erro desconhecido ';
```

```
End;
```

```
$$ LANGUAGE plpgsql;
```

Select * from produto;

Select retornaInfoProd(2);

Select retornaInfoProd(**100**);

Data Output		Explain
	retornainfoprod character varying	
1	Chocolate--6.27	

Data Output		Explain	Messages
	retornainfoprod character varying		
1	Nenhum produto encontrado		

A instrução RAISE

- A instrução **RAISE** é utilizada para gerar mensagens informativas e também para levantar exceções

RAISE nível 'formato' [, variável [, ...]];

RAISE NOTICE 'Chamando cs_create_job(%)', v_job_id;

```
FOR v_f IN vfilme(v_categoria) LOOP
    raise notice 'Título = %',v_f.titulo;
    raise notice 'Ano = %',v_f.ano;
END LOOP;
```

Exceção definida pelo desenvolvedor

RAISE nível 'formato' [, variável [, ...]];

**** O nível **EXCEPTION** causa um erro**

- Normalmente interrompe a transação corrente
- Os outros níveis apenas geram mensagens

RAISE EXCEPTION 'ID inexistente --> %', id_usuario;

**** É possível **capturar** esta exceção com EXCEPTION ... WHEN
RAISE_EXCEPTION THEN...**

Exemplo

```
create table  
logcontagem  
(id serial,  
info varchar(40));
```

```
CREATE OR REPLACE FUNCTION testaContagem(limite integer, vpais  
artista.pais%type)  
RETURNS void AS $$  
Declare  
    contagem integer;  
    info2 varchar(40);  
BEGIN  
    SELECT count(*) into contagem from artista where pais = vpais;  
    If contagem >= limite Then  
        RAISE exception 'Atingiu o limite';  
    End If;  
EXCEPTION  
    WHEN raise_exception THEN  
        info2 = 'Artistas de ' || vpais;  
        INSERT INTO logcontagem (id, info) VALUES (1, info2);  
END;  
$$ LANGUAGE plpgsql;
```

```
select testaContagem(5,'USA');  
select * from logcontagem;
```

```
82 select * from artista;  
83 update artista set pais = 'USA' where pais is null;  
84 insert into artista values(default,'Rosamund Pike',null,'USA', '27/01/1979');  
85
```

Data Output Explain Messages Notifications

	id integer	info character varying (40)
1	1	Artistas de USA possuem5
2	2	Artistas de USA possuem6

Exemplo

```
Create or replace function insereArttop(  
  cod IN top.codart%type,  
  nome IN top.nomeart%type,  
  cache1 IN top.cache%type)  
RETURNS void  
AS $$  
  
begin  
  insert into top values (cod,  
    nome,cache1);  
end;  
$$ LANGUAGE plpgsql;
```

```
select * from top;  
select insere_artista_top (17,'Teste',8);  
Select insere_artista_top(18,'Teste',6);
```

```
create or replace function insere_artista_top (vcod  
  artista.codart%type, vnome artista.nomeart%type,  
  nota numeric)  
Returns varchar as $$  
  Declare max_top integer;  
  vartop integer;  
Begin  
  max_top = 6;  
  select count(*) into vartop from top;  
  if vartop >= max_top then  
    raise exception 'top esgotada';  
  elsif nota > 7 then  
    perform insereArttop(vcod,vnome,null);  
    return 'Inclusão em Top com êxito!';  
  else  
    return 'Inclusão impossível';  
  end if;  
Exception  
  When raise_exception THEN  
    return 'Top Esgotada';  
  when others then  
    return 'Erro desconhecido';  
End;  
$$ LANGUAGE plpgsql;
```