



Banco de Dados II

Prof. Vinícius Alves Hax



Na aula anterior

- Programação dentro do banco de dados



Na aula de hoje


- Repetição básica
- Tipos dentro do PL/pgSQL

Repetição em PL/pgSQL

- Podemos ter uma estrutura do tipo 'repita' de três maneiras dentro do PL/pgSQL
 - Loop, for e while



Estrutura *loop*



```
do $$  
declare  
    counter int := 0;  
begin  
    loop  
        counter = counter + 1;  
        raise notice '%', counter;  
        exit when counter = 5;  
    end loop;  
end $$;
```

Loops com rótulos

-- Podemos adicionar um rótulo em um loop

do \$\$

declare

 counter int := 0;

begin

 <<my_loop>>

 loop

 counter = counter + 1;

 raise notice '%', counter;

 exit my_loop when counter = 5;

 end loop;

end \$\$;

Saindo de um loop alinhado

```
do $$
declare
  row_var int := 0;
  col_var int := 0;
begin
  <<outer_loop>>
  loop
    row_var = row_var + 1;
    <<inner_loop>>
    loop
      col_var = col_var + 1;
      raise notice '(%, %)', row_var, col_var;
      -- terminate the inner loop
      exit inner_loop when col_var = 3;
    end loop;
    -- reset the column
    col_var = 0;
    -- terminate the outer loop
    exit outer_loop when row_var = 3;
  end loop;
end $$;
```


Os tipos do PL/pgSQL

- Além do tipo integer (aula anterior) quais os tipos disponíveis?
 - Os mesmos tipos básicos do Postgres e mais alguns:
 - Ex: boolean, date, integer, real, text, time, varchar, etc

Exemplo

```
do $$  
declare  
    id integer = 1;  
    nome varchar(50) = 'John';  
    salario real = 2000.1;  
begin  
    raise notice 'O usuário com ID % e nome %  
recebeu % reais de salário', id, nome, salario;  
end $$;
```

Alteração de valores

- A alteração de valores de variáveis existentes pode ocorrer ao longo do programa, DESDE que as variáveis tenham sido declaradas
- Versões mais recentes aceitam tanto “:=” como somente “=” como comandos de atribuição

Exemplo

```
do $$
```

```
declare
```

```
    id integer;
```

```
    nome varchar(50) = 'John';
```

```
    salario real;
```


```
begin
```

```
    id := 1;
```


```
    salario = 2112.0; -- é possível misturar os sinais
```

```
    raise notice 'O usuário com ID % e nome % recebeu %  
    reais de salário', id, nome, salario;
```

```
end $$;
```




E se eu quiser utilizar os tipos de dados iguais
aos de alguma coluna de uma tabela?



E se eu quiser utilizar os tipos de dados iguais
aos de alguma coluna de uma tabela?

```
variavel nome_tabela.nome_coluna%type;
```



```
do $$  
declare  
    id student.teacher_id%type; -- Vai pegar o tipo daquela coluna  
begin  
    -- conta o número de registros  
    select teacher_id as contador  
    into id  
    from teacher where name = "Anakin";  
    -- mostra o resultado  
    raise notice 'Número é %', contador;  
end $$;
```

Tipo row (linha)

- DECLARE
 - Linha teacher%rowtype;
- ...

Tipo row

- DO \$\$
- DECLARE
- -- Declaramos uma variável de linha que terá o mesmo tipo de uma linha da tabela 'clientes'
- cliente_record clientes%ROWTYPE;
- BEGIN
- -- Usamos um loop FOR para iterar sobre cada linha da tabela 'clientes'
- FOR cliente_record IN SELECT * FROM clientes LOOP
- -- Exibimos os valores das colunas da linha atual
- RAISE NOTICE 'ID: %, Nome: %, Email: %', cliente_record.id, cliente_record.nome, cliente_record.email;
- END LOOP;
- END \$\$;

Referências

- Loop
<https://www.postgresqltutorial.com/postgresql-plpgsql/plpgsql-loop-statements/>
- Tipos de dados PostgreSQL
<https://www.postgresql.org/docs/current/datatype.html>