

# Trabalho 5

# Relatório

SCC-541 Laboratório de Bases de Dados

Leonardo Gonçalves Chahud - 5266649  
Murilo Franchi - 9790760  
Rafael Dantas - 12563686

Prof. Dr Caetano Traina Jr.  
PAE: Igor Alberte R. Eleutério

## Exercício 1

- Criar uma função que receba um nome de uma Escuderia (construtor) e retorne um valor tipo **TEXT** com o nome de sua nacionalidade.
- A função deve ser chamada **Nome\_Nacionalidade**.

```
create or replace function Nome_Nacionalidade(nome text)
  returns text as $$
  declare
    nacionalidade text;
  begin
    select into nacionalidade c.nationality
      from constructors c
      where c.name = nome;
    return nacionalidade;
  end;
$$ language plpgsql;
```

## Exercício 2

Criar uma função que receba uma nacionalidade e apresente todos os pilotos daquela nacionalidade usando mensagens no seguinte formato:

**<Contador> Nome: <Nome>**

Nesse formato:

- **<Contador>** é um inteiro que vai de 1 até a quantidade de pilotos encontrados;
- **<Nome>** é o nome completo do piloto (forename + surname).

A função deve se chamar **Pilotos\_Nacionalidade**.

```
create or replace function Pilotos_Nacionalidade(nacionalidade text)
returns void as $$
    declare piloto driver%rowtype;
    declare contador int;
begin
    contador := 1;
    for piloto in
        select *
            from driver d
            where d.nationality = nacionalidade
    loop
        raise notice '% : % %', contador, piloto.forename,
piloto.surname;
        contador := contador +1;
    end loop;
end;
$$ language plpgsql;
```

### Exercício 3

Criar um procedimento **Cidade Chamada** que receba o nome de uma cidade e apresente mensagens no seguinte formato:

**Contagem:** <contagem> |

**Nome:** <nome cidade>, **População:** <populacao>, **País:** <pais>

Onde:

- **<contagem>** é a quantidade de cidades encontradas com o nome informado;

As linhas seguintes representam cada uma das cidades encontradas e correspondem a:

- **<nome cidade>** é o nome de cada uma das cidades encontradas com o nome informado;
- **<populacao>** é a população de cada cidade;
- **<pais>** são seus respectivos países.

```
create or replace procedure Cidade_Chamada(cidade text)
  language plpgsql
  as $$
  declare contagem integer;
  declare nome_cidade geocities15k.name%Type := cidade;
  declare populacao geocities15k.population%Type;
  declare pais countries.name%Type;
  begin
    select into contagem count(*) over(partition by gc.name)
      from geocities15k gc
      where gc.name = cidade;
    raise notice 'Contagem: % |', contagem;

    for populacao, pais in
      select gc.population, c.name
      from geocities15k gc
      join countries c on c.code = gc.country and
gc.name = cidade
    loop
      raise notice 'Nome: %, População: %, País: %',
nome_cidade, populacao, pais;
    end loop;
  end; $$
```

## Exercício 4

Criar uma função chamada **Numero\_vitorias** que recebe três argumentos:

1. Nome do piloto;
  2. Sobrenome do piloto;
  3. Ano, como argumento opcional.
- A função deve executar uma consulta que retorna um número inteiro representando a quantidade de vezes em que o piloto foi campeão.
  - Caso o ano seja informado, as vitórias deverão ser restritas a apenas esse ano, e caso contrário, deve ser considerada toda a base de dados.

Dica: contar a quantidade de **Position = 1** na tabela **RESULTS**.

```
create or replace function Numero_vitorias(nome text, sobrenome text,
ano integer default null)
returns integer as $$
    declare total_vitorias integer;
    declare nome_completo text := nome || ' ' || sobrenome;
    begin
        case
            when ano is null then
                select into total_vitorias count(*)
                from driver d
                join results r on r.driverid = d.driverid
                and r.position = 1
                where d.forename = nome and d.surname =
sobrenome;

            when ano is not null then
                select into total_vitorias count(*)
                from driver d
                join results r on r.driverid = d.driverid
                and r.position = 1
                join races rc on rc.raceid = r.raceid and
rc.year = ano
                where d.forename = nome and d.surname =
sobrenome;
        end case;
        raise notice '% %', total_vitorias, nome_completo;
        return total_vitorias;
    end
    $$ language plpgsql;
```

## Exercício 5

Construa uma função **Pais\_Continente** que itere sobre a tabela **COUNTRIES** e retorne o **Nome** e o **Continente** dos países que tenham nomes com, no máximo, 15 caracteres.

- O resultado retornado deve ser de tipo **TABLE**.
- É obrigatório o uso de cursor.
- Depois de construída a função, execute o comando:  
**SELECT count(\*) FROM Pais \_Continente();**  
e informe a quantidade resultante no relatório.

Dica: Use a função **LENGTH** para avaliar o número de caracteres no nome.

```
CREATE OR REPLACE FUNCTION Pais_Continente()
  RETURNS TABLE (nome VARCHAR, continente VARCHAR) AS
$$
DECLARE
  cur CURSOR FOR SELECT name, continent FROM countries WHERE
length(name) <= 15;
  result RECORD;
BEGIN
  OPEN cur;

  LOOP
    FETCH cur INTO result;
    EXIT WHEN NOT FOUND;

    nome := result.name;
    continente := result.continent;
    RETURN NEXT;
  END LOOP;

  CLOSE cur;
END;
$$
LANGUAGE plpgsql;
```