



Universidade de São Paulo - São Carlos  
Instituto de Ciências Matemáticas e de Computação  
**SCC-541 Laboratório de Bases de Dados**  
**Trabalho Prático T6 – Exercícios sobre Visões (*views*)**

Dr. Caetano Traina Jr., Igor Alberte R. Eleutério  
1º semestre de 2023

**Data para entrega: 07 de junho**

## Datasets

### Base de dados da Fórmula 1 - FIA.

A base de dados a ser utilizada é a mesma preparada no **Trabalho 1**. Caso necessário, os arquivos originais para a carga de dados encontra-se no Google Drive, e podem ser acessados no endereço [este link](#). Para acessar os arquivos, você deve estar logado no Google Drive com sua conta USP. Cada arquivo corresponde a uma tabela, que deve ser carregada na base de dados.

Os dados originais foram ligeiramente modificados para facilitar o trabalho. Os originais podem ser obtidos nos seguintes *sites*:

- Dados da Fórmula-1: [Ergast Developer API](#)
- Países e Cidades do planeta: [GeoNames](#)
- Aeroportos: [OurAirports](#)

## Atividades da semana

- O objetivo deste trabalho é praticar o uso de **visões** (*views*) nos SGBDs.
- É obrigatória a criação de *views* em todos os exercício, exceto se a questão apontar o contrário.

---

**Exercício 1)** Crie uma visão materializada chamada [Aeroportos.Brasileiros](#) que contenha os Aeroportos das cidades brasileiras. Devem ser apresentados os dados do aeroporto (nome, latitude, longitude), os dados do país (nome e continente) e os dados da cidade vinculada (nome e população). Para fazer as junções, considere que o atributo [ISOCountry](#) de [Airports](#) se refere à [Code](#) em [Countries](#). Considere também que [Country](#), em [GeoCities15K](#), se refere à [Code](#) em [Countries](#). Por fim, considere que o atributo [City](#) em [Airports](#) se refere à [Name](#) em [GeoCities15K](#). Apresente o número de tuplas dessa visão no relatório.

(a) Discuta as vantagens e desvantagens de se utilizar visões materializadas.

---

**Exercício 2)** Com base na questão anterior, elabore uma visão que apresente apenas os aeroportos brasileiros que não estejam vinculados a nenhuma cidade. Nomeie essa visão como [Aeroportos\\_sem\\_cidades](#).

**Dica:** Uma abordagem é utilizar a cláusula [EXCEPT](#).

Depois, crie uma visão que contenha apenas as cidades brasileiras com população de no

mínimo 100 mil habitantes. Essa visão deve se chamar `Cidades_brasileiras`. Por fim, associe a cada aeroporto de `Aeroportos_sem_cidades` as cidades brasileiras da visão `Cidades_brasileiras` que estejam a no máximo 10Km. Apresente o nome do aeroporto, o nome e a população da cidade, e a distância entre o aeroporto e a cidade. Note que, para cada aeroporto, podem ser encontradas várias cidades.

Deve ser definida uma função para calcular as distâncias entre pontos na superfície da Terra. É possível definir uma, ou pode ser usada a função `EarthDistance` disponível nas `EXTENSIONS` `EarthDistance` e `CUBE`, tal como exemplificado a seguir:

```
CREATE EXTENSION IF NOT EXISTS Cube;
CREATE EXTENSION IF NOT EXISTS EarthDistance; -- Inclui EarthDistance(),
-- precisa de Cube

SELECT A.Name, C.Name, EarthDistance(LL_to_Earth(A.Lat , A.Long),
                                     LL_to_Earth(C.Lat , C.Long) )
FROM (SELECT Name, LatDeg lat, LongDeg Long
      FROM Airports
      WHERE Name ~*'Guarulhos' AND Type='Large_airport'
      ) A,
      (SELECT Name , Lat , Lng Long
      FROM Circuits
      WHERE CircuitRef ~*'Interlagos '
      ) C;
```

Alguns pontos importantes sobre esse código:

- O comando calcula a distância entre o aeroporto cujo nome se inicia por `Guarulhos` e que seja do tipo `large_airport` e o circuito que tem o nome de referência `Interlagos` (desconsiderando a caixa da letra).
- Os comandos `CREATE EXTENSION` nas linhas 1 e 2 instalam as extensões necessárias usar as funções de cálculo de distância.
- A função `LL_to_Earth` retorna a localização de um ponto na superfície terrestre a partir de uma Latitude (argumento 1) e Longitude (argumento 2).
- A função `EarthDistance` retorna a distância entre dois pontos na superfície terrestre.
- O operador `*` compara duas strings, considerando expressões regulares, sem considerar a caixa da letra.

Para uma descrição mais detalhada, consulte a documentação, disponível em:

<https://www.postgresql.org/docs/current/earthdistance.html>.

---

**Exercício 3)** Crie uma visão chamada `Circuitos_completa` que apresente informações dos circuitos e de seus respectivos países. Devem ser apresentados os atributos: `Name`, `Location` e `Country` da tabela `Circuits` e os atributos `Code` e `Continent` da tabela `Countries`. Essa visão deve apresentar todos os circuitos, independentemente se foram encontrados os respectivos países ou não. Considere que o atributo `Name` em `Countries` corresponde ao atributo `Country` em `Circuits`. Apresente o número de tuplas dessa visão no relatório.

---

**Exercício 4)** Considerando a questão anterior, crie uma visão chamada `Problemas_circuitos` que apresente apenas os circuitos para os quais não foram encon-

trados os respectivos países na tabela **Countries**.

**Dica:** uma possibilidade para resolver a esta questão é usar a cláusula **EXCEPT**.

- (a) Por que houve problema ao tentar encontrar os países desses circuitos na tabela **Countries**?

```
SELECT * FROM problemas_circuitos;
```

Não existem os países **Korea**, **UK**, **UAE**, **USA** na tabela **COUNTRIES**.

---

**Exercício 5)** Crie uma visão chamada **Correção\_circuitos** que liste apenas os circuitos vinculados aos problemas identificados. Essa visão deve apresentar os atributos **Name**, **Location** e **Country** da tabela **Circuits**. Não deve ser executada nenhuma junção com a tabela **Countries** para esta questão.

Depois, atualize os nomes dos países nos circuitos utilizando a visão criada a fim de haver uma ligação com a tabela **Countries**. Execute os comandos e mostre o resultado da execução, junto com o número de tuplas afetadas, para cada uma das correções necessárias.

## Entrega

Cada equipe deverá entregar dois arquivos no **Escaninho** do **Tidia**, sendo que apenas o líder de cada equipe deverá colocar, no seu *Escaninho*, em uma pasta com o nome **T5**:

- Um arquivo com o script no formato **.zip**, contendo um arquivo **.sql** com os comandos **SQL** utilizados para cada atividade.
- Um arquivo com o relatório SUCINTO no formato **.pdf**.
- O relatório deverá apresentar testes (pelo menos um por questão) para cada um dos procedimentos/funções criados em cada questão, com uma captura de tela do resultado para cada teste.
- Quando os resultados forem muito longos, o grupo deverá apresentar somente as primeiras tuplas (por exemplo, as 10 primeiras linhas de uma tabela resultante).

Como a atividade tem prazo de entrega máximo de uma hora antes da próxima aula, os arquivos devem ser submetidos

até às 18h00 do dia 31 de maio, *com postagem somente do líder da equipe*.

Não serão aceitos projetos feitos à mão e a organização clara das respostas também é um ponto avaliado.

Plágio será avaliado com nota zero.

Bom Trabalho!