

Trabalho 7

Relatório

SCC-541 Laboratório de Bases de Dados

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

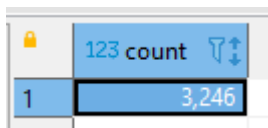
Prof. Dr Caetano Traina Jr.
PAE: Igor Alberte R. Eleutério

Exercício 1

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.

```
create materialized view aeroportos_brasileiros as
select c.continent, c.name as country_name,
       gc.name as city_name, gc.population,
       a.name as airport_name, a.latdeg as lat, a.longdeg as
lon
       from countries c
       join geocities15k gc on c.code = gc.country
       join airports a on a.city = gc.name
       where a.isocountry = 'BR'
       order by gc.name;
```

```
select count(*) from aeroportos_brasileiros;
```



123 count
1 3,246

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

A vantagem de usar uma visão materializada é que essa visão é armazenada no banco de dados como uma tabela, caso precise dos dados dessa visão, ela não precisa ser “recalculada” toda vez que é chamada. Entretanto, como a visão é armazenada, ela ocupa espaço no banco de dados, o que pode ser custoso caso a visão seja muito “grande”.

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 **Earth_Distance** 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, Earth_Distance(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 para 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 **Earth_Distance** 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>.

```
create materialized view aeroportos_sem_cidade as
  select c.continent, c.name as country_name,
         a.name as airport_name, a.latdeg as lat, a.longdeg as
lon
         from countries c
         join airports a on a.isocountry = c.code
         where c.code = 'BR' and a.city is null;
```

```
create materialized view cidades_brasileiras as
  select gc.name as city_name, gc.population
         from geocities15k gc
         where gc.population >= 100000 and gc.country = 'BR';
```

```
CREATE EXTENSION IF NOT EXISTS Cube;
CREATE EXTENSION IF NOT EXISTS EarthDistance;


select a.airport_name, c.city_name, c.population,
       earth_distance(ll_to_earth(a.lat, a.lon), ll_to_earth(c.lat,
c.lon)) as dist
  from aeroportos_sem_cidade a, cidades_brasileiras c
  where earth_distance(ll_to_earth(a.lat, a.lon), ll_to_earth(c.lat,
c.lon)) <= 10000;
```

	ABC airport_name	ABC city_name	123 population	123 dist
1	Guarulhos - Sao Paulo Airport Marriott Heliport	Guarulhos	1,169,577	4,313.1482922706

Exercício 3

Crie uma visão chamada **Circuitos_completa** que apresenta 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.

```
create or replace view circuitos_completa as
    select c.name as circuit_name, c.location as circuit_location,
    c.country,
           ct.code as country_code, ct.continent
    from circuits c
    left join countries ct on c.country = ct.name
    order by c.name;
```

	123 count 
1	77

Exercício 4

Considerando a questão anterior, crie uma visão chamada **Problemas_circuitos** que apresente apenas os circuitos para os quais não foram encontrados os respectivos países na tabela **Countries**.

Dica: uma possibilidade para resolver 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 países com nome **Korea**, **UK**, **UAE**, **USA** na tabela **COUNTRIES**.

```
create or replace view problemas_circuitos as
  select c.circuit_name, c.circuit_location, c.country
  from circuitos_completa c
  where c.country_code is null;
```

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.

```
create or replace view correcao_circuitos as
    select c.name as circuit_name, c.location as circuit_location,
    c.country
        from circuits c
        where c.country = 'UK' or c.country = 'USA' or c.country =
'Korea' or c.country = 'UAE';
```

Antes:

	ABC circuit_name	ABC circuit_location	ABC country
1	Aintree	Liverpool	UK
2	Brands Hatch	Kent	UK
3	Circuit of the Americas	Austin	USA
4	Detroit Street Circuit	Detroit	USA
5	Donington Park	Castle Donington	UK
6	Fair Park	Dallas	USA
7	Indianapolis Motor Speedway	Indianapolis	USA
8	Korean International Circuit	Yeongam County	Korea
9	Las Vegas Street Circuit	Nevada	USA
10	Long Beach	California	USA
11	Miami International Autodrome	Miami	USA
12	Phoenix street circuit	Phoenix	USA
13	Riverside International Raceway	California	USA
14	Sebring International Raceway	Florida	USA
15	Silverstone Circuit	Silverstone	UK
16	Watkins Glen	New York State	USA
17	Yas Marina Circuit	Abu Dhabi	UAE

Correções:

UK se refere ao país United Kingdom na tabela **Countries**. Antes da correção temos:

```
select * from correcao_circuitos
where country = 'UK';
```

	circuit_name text	circuit_location text	country text
1	Silverstone Circuit	Silverstone	UK
2	Donington Park	Castle Donington	UK
3	Brands Hatch	Kent	UK
4	Aintree	Liverpool	UK

Aplicando a correção:

```
update correcao_circuitos
set country = 'United Kingdom'
where country = 'UK';
```

Atualizamos 4 tuplas:

Data Output	Messages	Notifications
UPDATE 4		
Query returned successfully in 95 msec.		

e agora **Circuits** possui:

```
select * from circuitos_completa
where country = 'United Kingdom'
```

	circuit_name text	circuit_location text	country text	country_code character (2)	continent character (2)
1	Aintree	Liverpool	United Kingdom	GB	EU
2	Brands Hatch	Kent	United Kingdom	GB	EU
3	Donington Park	Castle Donington	United Kingdom	GB	EU
4	Silverstone Circuit	Silverstone	United Kingdom	GB	EU

Fazemos o mesmo para USA, que se refere a United States:

Antes da Correção:

```
select * from correcao_circuitos
where country = 'USA';
```

	circuit_name text	circuit_location text	country text
1	Indianapolis Motor Speedway	Indianapolis	USA
2	Phoenix street circuit	Phoenix	USA
3	Detroit Street Circuit	Detroit	USA
4	Fair Park	Dallas	USA
5	Long Beach	California	USA
6	Las Vegas Street Circuit	Nevada	USA
7	Watkins Glen	New York State	USA
8	Riverside International Raceway	California	USA
9	Sebring International Raceway	Florida	USA
10	Circuit of the Americas	Austin	USA
11	Miami International Autodrome	Miami	USA

Aplicando a correção:

```
update correcao_circuitos
set country = 'United States'
where country = 'USA';
```

Data Output Messages Notifications

UPDATE 11

Query returned successfully in 83 msec.

e após a correção:

	circuit_name text	circuit_location text	country text	country_code character (2)	continent character (2)
1	Circuit of the Americas	Austin	United States	US	NA
2	Detroit Street Circuit	Detroit	United States	US	NA
3	Fair Park	Dallas	United States	US	NA
4	Indianapolis Motor Speedway	Indianapolis	United States	US	NA
5	Las Vegas Street Circuit	Nevada	United States	US	NA
6	Las Vegas Strip Street Circuit	Las Vegas	United States	US	NA
7	Long Beach	California	United States	US	NA
8	Miami International Autodrome	Miami	United States	US	NA
9	Phoenix street circuit	Phoenix	United States	US	NA
10	Riverside International Raceway	California	United States	US	NA
11	Sebring International Raceway	Florida	United States	US	NA
12	Watkins Glen	New York State	United States	US	NA

Para Korea, que se refere a South Korea na tabela **Countries**:

	circuit_name text	circuit_location text	country text
1	Korean International Circuit	Yeongam County	Korea

Correção:

```
update correcao_circuitos
set country = 'South Korea'
where country = 'Korea';
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 92 msec.

Após Correção:

	circuit_name text	circuit_location text	country text	country_code character (2)	continent character (2)
1	Korean International Circuit	Yeongam County	South Korea	KR	AS

E finalmente UAE se refere a United Arab Emirates na tabela **country**.

	circuit_name text	circuit_location text	country text
1	Yas Marina Circuit	Abu Dhabi	UAE

Correção:

```
update correcao_circuitos
set country = 'United Arab Emirates'
where country = 'UAE';
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 90 msec.

Após a correção:

	circuit_name text	circuit_location text	country text	country_code character (2)	continent character (2)
1	Yas Marina Circuit	Abu Dhabi	United Arab Emirates	AE	AS