

Tarefa02

February 21, 2021

```
[1]: %LOAD postgresql dbname=Filmes
```

0.1 Script SQL para a Tarefa 02 - Banco de Dados II 2020.2

0.1.1 21/02/2021

0.1.2 Patric Lacouth - 20191370047

0.1.3 1. Faça inicialmente a seguinte inserção:

```
insert into filme values(default,'Superman',2018,120,null,3);
```

```
[4]: INSERT INTO
      filme(codfilme,titulo,ano,duracao,codcateg,codest)
VALUES
      (default,'Superman',2018,120,null,3);
```

0.1.4 2. Verifique quais os títulos dos filmes que possuem duração maior que 120 min (0,5)?

```
[6]: SELECT * FROM filme WHERE(duracao > 120);
```

```
[6]: +-----+-----+-----+-----+-----+-----+
| codfilme | titulo           | ano  | duracao | codcateg | codest |
+-----+-----+-----+-----+-----+-----+
| 1        | Encontro Explosivo | 2010 | 134     | 4        | 1       |
+-----+-----+-----+-----+-----+-----+
| 2        | O Besouro Verde   | 2010 | 155     | 1        | 1       |
+-----+-----+-----+-----+-----+-----+
| 3        | Comer, Rezar, Amar | 2010 | 177     | 2        | 1       |
+-----+-----+-----+-----+-----+-----+
| 4        | Coringa           | 2019 | 122     | 6        | 1       |
+-----+-----+-----+-----+-----+-----+
| 8        | The Matrix        | 1999 | 130     | 4        | 5       |
+-----+-----+-----+-----+-----+-----+
| 9        | Clube da Luta     | 1999 | 140     | 4        | 1       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0,00 sec)
```

0.1.5 3. Na tabela ARTISTA, quais artistas possuem cidade nula? Após a consulta, atualize as cidades nulas encontradas para três artistas (0,5).

```
[7]: SELECT * FROM artista WHERE(cidade IS NULL);
```

```
[7]: +-----+-----+-----+-----+-----+
| codart | nomeart      | cidade | pais | datanasc      |
+-----+-----+-----+-----+-----+
| 1      | Cameron Diaz | NULL   | USA  | Tue Jul 15 00:00:00 1975 |
+-----+-----+-----+-----+-----+
| 2      | Julia Roberts | NULL   | USA  | Sat Aug 20 00:00:00 2067 |
+-----+-----+-----+-----+-----+
| 5      | Bradley Cooper | NULL   | USA  | Mon Jun 6 00:00:00 1977 |
+-----+-----+-----+-----+-----+
| 6      | Tom Cruise    | NULL   | USA  | Wed Sep 10 00:00:00 2064 |
+-----+-----+-----+-----+-----+
| 3      | Brad Pitt     | NULL   | USA  | Thu Mar 5 00:00:00 1970 |
+-----+-----+-----+-----+-----+
| 4      | Joaquin Phoenix | NULL   | USA  | Thu Apr 6 00:00:00 1972 |
+-----+-----+-----+-----+-----+
6 rows in set (0,00 sec)
```

0.1.6 4. Qual a descrição da categoria do filme ‘Coringa’ (1,0)?

```
[8]: SELECT c.desccateg AS descrição FROM
      filme f JOIN categoria c ON f.codcateg = c.codcateg
      WHERE(f.titulo = 'Coringa');
```

```
[8]: +-----+
| descrição |
+-----+
| Drama     |
+-----+
1 row in set (0,00 sec)
```

0.1.7 5. Mostre os títulos de filmes, seus estúdios e suas categorias (1,0).

```
[21]: SELECT f.titulo as título, e.nomeest as estúdio, c.desccateg FROM
      filme f JOIN estudio e ON e.codest = f.codest
      LEFT JOIN categoria c ON f.codcateg = c.codcateg;
```

```
[21]: +-----+-----+-----+
| título          | estúdio  | desccateg |
+-----+-----+-----+
| Encontro Explosivo | Paramount | Ação      |
+-----+-----+-----+
| O Besouro Verde   | Paramount | Aventura  |
+-----+-----+-----+
```

```

+-----+-----+-----+
| Comer, Rezar, Amar      | Paramount | Romance  |
+-----+-----+-----+
| Coringa                  | Paramount | Drama    |
+-----+-----+-----+
| Era uma vez em Hollywood | Disney    | Ação     |
+-----+-----+-----+
| Nasce uma estrela        | Paramount | Drama    |
+-----+-----+-----+
| PI                       | Paramount | Drama    |
+-----+-----+-----+
| The Matrix               | Warner    | Ação     |
+-----+-----+-----+
| Clube da Luta            | Paramount | Ação     |
+-----+-----+-----+
| Superman                 | Universal | NULL     |
+-----+-----+-----+
10 rows in set (0,00 sec)

```

0.1.8 6. Qual o nome dos artistas que fizeram o filme ‘Encontro Explosivo’ (1,0)?

```

[34]: SELECT a.nomeart AS nome FROM
        artista a JOIN personagem p ON a.codart = p.codart
        JOIN filme f ON f.codfilme = p.codfilme
        WHERE(f.titulo = 'Encontro Explosivo');

```

```

[34]: +-----+
| nome      |
+-----+
| Cameron Diaz |
+-----+
| Julia Roberts |
+-----+
2 rows in set (0,00 sec)

```

0.1.9 7. Selecione os artistas que têm o nome iniciando com a letra ‘B’ e participaram de filmes da categoria ‘Aventura’ (1,0).

```

[45]: SELECT * FROM
        artista a JOIN personagem p ON a.codart = p.codart
        JOIN filme f ON p.codfilme = f.codfilme
        JOIN categoria c ON c.codcateg = f.codcateg
        WHERE( (a.nomeart LIKE 'B%') AND (c.desccateg = 'Aventura') );

```

```

[45]: Empty set (0,00 sec)

```

0.1.10 8. Apresente quantos filmes existem por categoria. Para isso mostre a descrição da categoria e sua respectiva contagem (1,0).

```
[54]: SELECT c.desccateg, COUNT(*) FROM
      filme f JOIN categoria c ON c.codcateg = f.codcateg
      GROUP BY c.desccateg;
```

```
[54]: +-----+-----+
      | desccateg | count |
      +-----+-----+
      | Ação      | 4      |
      +-----+-----+
      | Romance    | 1      |
      +-----+-----+
      | Aventura    | 1      |
      +-----+-----+
      | Drama      | 3      |
      +-----+-----+
      4 rows in set (0,00 sec)
```

0.1.11 9. Altere a questão anterior acrescentando a cláusula “having”. Explique o que a consulta retorna no contexto que você implementou (1,0)

```
[62]: SELECT c.desccateg, COUNT(*) AS Total FROM
      filme f JOIN categoria c ON c.codcateg = f.codcateg
      GROUP BY c.desccateg
      HAVING COUNT(*) > 1;
```

```
[62]: +-----+-----+
      | desccateg | total |
      +-----+-----+
      | Ação      | 4      |
      +-----+-----+
      | Drama      | 3      |
      +-----+-----+
      2 rows in set (0,00 sec)
```

A cláusula HAVING permite filtrar o resultado obtido pelo GROUP BY para obter apenas as categorias associadas com mais de um filme no banco de dados.

0.1.12 10. O que o seguinte comando mostra (1,0)?

```
select a.nomeart, p.nomepers
from artista a left outer join personagem p on a.codart = p.codart;
```

```
[64]: SELECT a.nomeart, p.nomepers FROM
      artista a LEFT OUTER JOIN personagem p ON a.codart = p.codart;
```

```
[64]: +-----+-----+
      | nomeart      | nomepers  |
      +-----+-----+
      | Cameron Diaz  | Natalie   |
      +-----+-----+
      | Julia Roberts  | Tom       |
      +-----+-----+
      | Brad Pitt     | John      |
      +-----+-----+
      | Julia Roberts  | Ana       |
      +-----+-----+
      | Bradley Cooper | Tom       |
      +-----+-----+
      | Joaquin Phoenix | John      |
      +-----+-----+
      | Fulano de Tal  | Neo       |
      +-----+-----+
      | Rick Martinez  | Rick      |
      +-----+-----+
      | Brad Pitt     | Tyler Durden |
      +-----+-----+
      | Alessandra Negrini | NULL      |
      +-----+-----+
      | Fernanda Montenegro | NULL      |
      +-----+-----+
      | Tom Cruise     | NULL      |
      +-----+-----+
      | Rodrigo Santoro  | NULL      |
      +-----+-----+
      13 rows in set (0,00 sec)
```

O comando mostra o nome de todos os artistas relacionado com seus personagens, mesmo que o artista não tenha nenhum personagem associado.

0.1.13 11. O que a seguinte consulta retorna (1,0)?

```
Select f.titulo as Filme, c.desccateg as Categoria
From filme f right join categoria c on f.codcateg = c.codcateg
Where c.codcateg is null;
```

```
[70]: SELECT f.titulo AS Filme, c.desccateg AS Categoria
      FROM filme f RIGHT JOIN categoria c ON f.codcateg = c.codcateg
      WHERE c.codcateg IS NULL;
```

```
[70]: Empty set (0,00 sec)
```

A consulta retorna uma tabela vazia por causa do RIGHT JOIN que impede que o registro que

possua categoria NULL seja apresentado, uma vez que a tabela que contém o NULL está a direita da consulta. Abaixo está o resultado da consulta com LEFT JOIN.

```
[71]: SELECT f.titulo AS Filme, c.desccateg AS Categoria
      FROM filme f LEFT JOIN categoria c ON f.codcateg = c.codcateg
      WHERE c.codcateg IS NULL;
```

```
[71]: +-----+-----+
      | filme   | categoria |
      +-----+-----+
      | Superman | NULL      |
      +-----+-----+
      1 row in set (0,00 sec)
```

0.1.14 12. Crie uma consulta sua ao banco Filmes. Use operadores SQL e algum tipo de JOIN. Formule o enunciado da consulta e mostre sua solução em SQL (1,0).

Qual o total gasto em cache por cada estúdio?

```
[80]: SELECT e.nomeest AS estudio, SUM(p.cache) AS cache FROM
      filme f JOIN personagem p ON f.codfilme = p.codfilme
      JOIN estudio e ON f.codest = e.codest
      GROUP BY e.nomeest
      ORDER BY SUM(p.cache) DESC
```

```
[80]: +-----+-----+
      | estudio  | cache    |
      +-----+-----+
      | Paramount | 286465,  |
      +-----+-----+
      | Warner    | 172500,  |
      +-----+-----+
      | Disney    | 11500,   |
      +-----+-----+
      3 rows in set (0,00 sec)
```