

HO08: SQL (DML)

Aluno: Bernardo Ladeira Borges Kartabil

Matrícula: 838966

Turno: Manhã

QUESTÃO 1:

Projetar o primeiro nome e o último nome dos atores que são diretores;

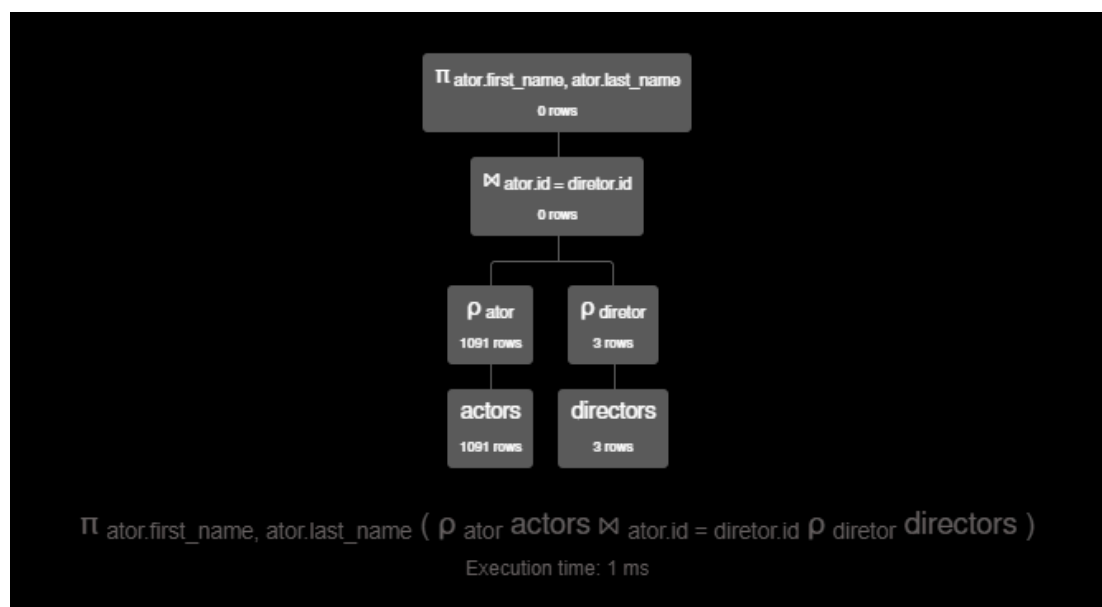
```
SELECT ator.first_name, ator.last_name  
FROM actors AS ator  
JOIN directors AS diretor ON ator.id = diretor.id;
```



The screenshot shows a SQL query editor with a dark theme. At the top, there is a tab labeled "select from where group having order limit". The query is entered in the main text area and is as follows:

```
1 SELECT ator.first_name, ator.last_name  
2 FROM actors AS ator  
3 JOIN directors AS diretor ON ator.id = diretor.id;
```

At the bottom of the editor, there is a blue button with a play icon and the text "Executar consulta". To the right of this button, there are two links: "Download" with a download icon and "Histórico" with a circular arrow icon.



Query execution plan diagram showing a join between actors and directors tables, followed by a projection of actor names. The join condition is ator.id = diretor.id. The actors table has 1091 rows and the directors table has 3 rows. The join result has 0 rows, and the final projection also has 0 rows.

Execution time: 1 ms

ator.first_name	ator.last_name
-----------------	----------------

QUESTÃO 2:

Projetar o primeiro nome e o último nome dos atores que não são diretores;

Professor, fiz essa questão duas vezes, pois percebi que, da forma com que fiz a consulta na primeira vez, o programa fez um produto cartesiano das tabelas de ator e diretor, tornando-se algo custoso.

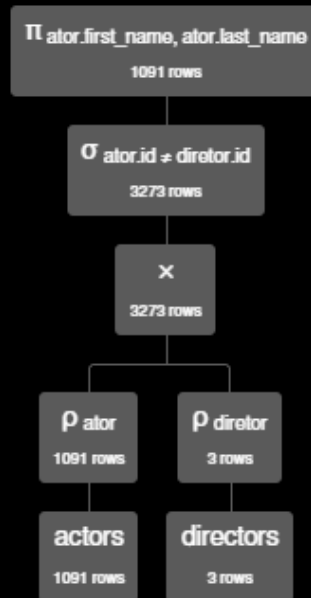
PRIMEIRA FORMA:

-- Projetar o primeiro nome e o último nome dos atores que não são diretores;

```
SELECT ator.first_name, ator.last_name  
FROM actors AS ator , directors AS diretor  
WHERE ator.id != diretor.id;
```

select from where group having order limit

```
1 -- Projetar o primeiro nome e o último nome dos atores que não são diretores;  
2  
3 SELECT ator.first_name, ator.last_name  
4 FROM actors AS ator, directors AS diretor  
5 WHERE ator.id != diretor.id
```



$\Pi_{\text{ator.first_name}, \text{ator.last_name}} \sigma_{\text{ator.id} \neq \text{diretor.id}} (\rho_{\text{ator}} \text{actors} \times \rho_{\text{diretor}} \text{directors})$

Execution time: 1 ms

ator.first_name	ator.last_name
'Chris'	'Anastasio'
'Michael'	'Beach'
'John'	'Bedford Lloyd'
'Michael'	'Biehn'
'Captain Kidd'	'Brewer Jr.'
'Leo'	'Burmester'
'Mike (I)'	'Cameron'
'J. Kenneth'	'Campbell'
'Michael (I)'	'Chapman'
'Phillip'	'Darlington'

« 1 2 3 »

SEGUNDA FORMA:

/ leva as colunas preenchidas de diretores para a tabela de atores se "ator.id = diretor.id", se não, leva as informações de diretores como null*/*

```
SELECT ator.first_name, ator.last_name
FROM actors AS ator
LEFT JOIN directors AS diretor ON ator.id = diretor.id
WHERE diretor.id IS NULL
```

select from where group having order limit

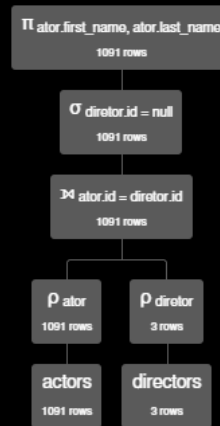
```

1 -- Projetar o primeiro nome e o último nome dos atores que não são diretores;
2
3 /* leva as colunas preenchidas de diretores para a tabela de atores se "ator.id = diretor.id", se não, leva as informações
4 de diretores como null*/
5
6 SELECT ator.first_name, ator.last_name
7 FROM actors AS ator
8 LEFT JOIN directors AS diretor ON ator.id = diretor.id
9 WHERE diretor.id IS NULL

```

▶ Executar consulta

Download Histórico



$\pi_{\text{ator.first_name, ator.last_name}} \sigma_{\text{diretor.id = null}} (\rho_{\text{ator}} \text{actors} \bowtie_{\text{ator.id = diretor.id}} \rho_{\text{diretor}} \text{directors})$

Execution time: 0 ms

```
 $\Pi$  ator.first_name, ator.last_name  $\sigma$  diretor.id = null (  $\rho$  ator actors  $\bowtie$  ator.id = diretor.id  $\rho$  diretor directors )
```

Execution time: 0 ms

ator.first_name	ator.last_name
'Chris'	'Anastasio'
'Michael'	'Beach'
'John'	'Bedford Lloyd'
'Michael'	'Biehn'
'Captain Kidd'	'Brewer Jr.'
'Leo'	'Burmester'
'Mike (I)'	'Cameron'
'J. Kenneth'	'Campbell'
'Michael (I)'	'Chapman'
'Phillip'	'Darlington'

OBSERVAÇÃO:

Como podemos notar, nessa segunda forma, a quantidade de linhas para verificar foi de 3273 para 1091. Estou começando a me preocupar com o custo de minha consulta.

QUESTÃO 3:

Projetar o primeiro nome e o último nome dos atores e diretores;

OBSERVAÇÃO:

Eu não sei por que a calculadora só mostrou os nomes dos atores e não os de ambas as tabelas.

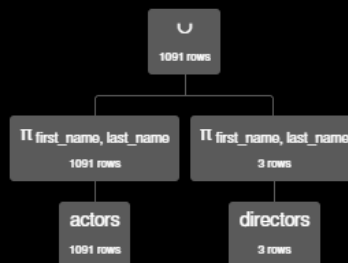
```
SELECT first_name, last_name
FROM actors
UNION
SELECT first_name, last_name
FROM directors;
```

select from where group having order limit

```
1 -- Projetar o primeiro nome e o último nome dos atores e diretores;
2
3 SELECT first_name, last_name
4 FROM actors
5 UNION
6 SELECT first_name, last_name
7 FROM directors;
```

▶ Executar consulta

Download Histórico



$\pi_{\text{first_name, last_name}} \text{actors} \cup \pi_{\text{first_name, last_name}} \text{directors}$
Execution time: 0 ms

1091 rows 3 rows

π first_name, last_name actors \cup π first_name, last_name directors

Execution time: 0 ms

actors.first_name	actors.last_name
'Chris'	'Anastasio'
'Michael'	'Beach'
'John'	'Bedford Lloyd'
'Michael'	'Biehn'
'Captain Kidd'	'Brewer Jr.'
'Leo'	'Burmester'
'Mike (I)'	'Cameron'
'J. Kenneth'	'Campbell'
'Michael (I)'	'Chapman'
'Phillip'	'Darlington'

< 1 2 3 >

QUESTÃO 4:

Projetar o nome dos filmes que não são dirigidos por nenhum diretor;

-- *Projetar o nome dos filmes que não são dirigidos por nenhum diretor;*

SELECT name FROM movies AS filmes

JOIN movies_directors AS diretores_filmes ON diretores_filmes.movie_id = filmes.id

WHERE director_id IS NULL;

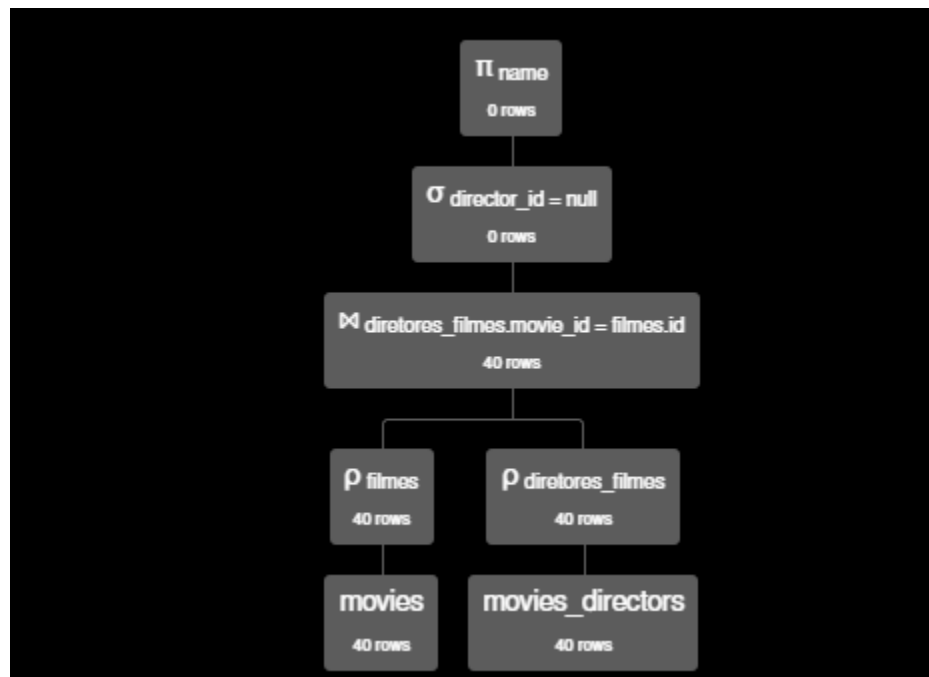
select from where group having order limit

```
1 -- Projetar o nome dos filmes que não são dirigidos por nenhum diretor;
2
3 SELECT name FROM movies AS filmes
4 JOIN movies_directors AS diretores_filmes ON diretores_filmes.movie_id = filmes.id
5 WHERE director_id IS NULL
```

▶ Executar consulta

Download

Histórico



$\Pi_{name} \sigma_{director_id = null} (\rho_{filmes} \text{ movies } \bowtie_{diretores_filmes.movie_id = filmes.id} \rho_{diretores_filmes} \text{ movies_directors})$

Execution time: 0 ms

filmes.name

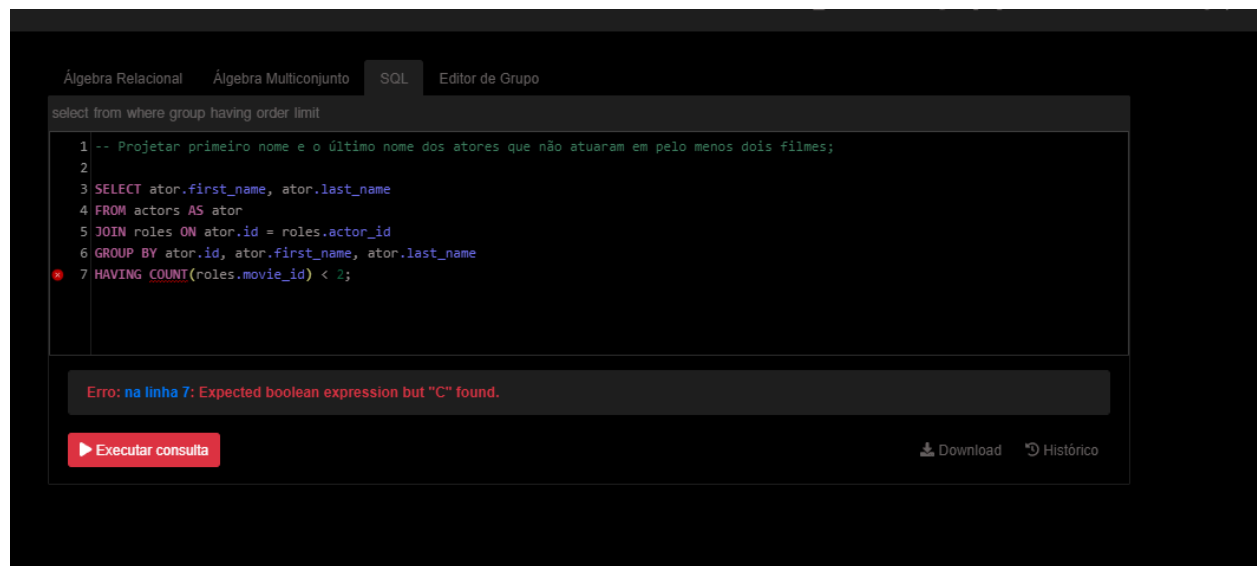
< >

QUESTÃO 5:

Observação: A calculadora apresentou erro

Projetar primeiro nome e o último nome dos atores que não atuaram em pelo menos dois filmes;

```
SELECT ator.id, ator.first_name, ator.last_name
FROM actors AS ator
JOIN roles ON ator.id = roles.actor_id
GROUP BY ator.id, ator.first_name, ator.last_name
HAVING COUNT(*) < 2;
```



QUESTÃO 6:

Tentei fazer essa questão, porém, além de difícil, ela simplesmente não funcionou por causa do COUNT.