



PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS

DML: SELECT



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

eliezio.soares@ifrn.edu.br | <https://docente.ifrn.edu.br/elieziosoares>

Msc. Eliezio Soares

COMANDO SELECT - QUERY

- O comando responsável por obter dados em um banco de dados é chamado **query**.
- A sintaxe de uma query é composta basicamente por:

```
SELECT Lista_seleção FROM Expressão_Table [Filtros] [Ordenação] [...]
```

- Lista_seleção: Lista de colunas selecionadas;
- Expressão_Table: Tabela a ser consultada pela query;
- Filtros: Critérios para seleção dos dados da tabela;
- Ordenação: Ordem de apresentação dos dados selecionados;

COMANDO SELECT - QUERY

- O comando SELECT, seleciona dados e os retorna.
- Exemplos:

```
SELECT 4 *5;
```

```
SELECT 'EXEMPLO QUERY';
```

```
SELECT * FROM ATOR;
```

SELEÇÃO DE CAMPOS

- A lista de seleção pode ser:
 - Um subconjunto das colunas disponíveis na tabela;
 - Cálculos sobre um subconjunto de campos;
 - * - Tudo

```
SELECT titulo, descrição FROM Filme
```

```
SELECT titulo, descrição, (2 * taxa) FROM Filme
```

```
SELECT * FROM Filme
```

```
SELECT *, (2 * taxa) FROM Filme
```

EXPRESSÃO TABLE

- A expressão TABLE computa uma tabela e **contém**:
 - Uma cláusula **FROM**;
- Pode conter, opcionalmente:
 - WHERE
 - GROUP BY
 - HAVING
 - ORDER BY

Filtros,
ordenação,
Agrupamentos

```
SELECT * FROM ATOR WHERE ator_id = 5;
```

EXPRESSÃO TABLE FROM

- A cláusula *FROM* especifica:
 - Uma tabela;
 - Uma subquery;
 - Um JOIN (junção de tabelas);
 - Combinações das opções anteriores...

```
SELECT * FROM ATOR;
```

```
SELECT * FROM ATOR WHERE ator_id = 5;
```

```
SELECT * FROM ATOR ORDER BY nome;
```

EXPRESSÃO TABLE

ORDER BY

- A cláusula ORDER BY é utilizada para ordenar a output table a partir dos valores de uma ou mais colunas.

```
SELECT * FROM ATOR ORDER BY nome;
```

- Por padrão o ORDER BY ordena os registros em ordem **ASC**endente.
- Para ordenar em ordem **DESC**endente, utilize o termo **DESC**.

```
SELECT * FROM ATOR ORDER BY nome desc;
```

```
SELECT * FROM ATOR ORDER BY nome asc;
```

EXPRESSÃO TABLE

WHERE

- **Where** define uma condição:
 - Qualquer expressão que retorne um valor do tipo booleano.
- Cada linha da tabela virtual gerada pela cláusula **FROM**, é checada de acordo com a condição **WHERE**.
- Se o resultado for **TRUE**, a linha será considerada na tabela de saída (output table).
- Se o resultado for **FALSE** ou **NULL**, a linha é descartada.
- A condição **WHERE**, normalmente referencia uma coluna da tabela gerada pela query.

EXPRESSÃO TABLE

WHERE

- Exemplos de sentenças WHERE:

```
SELECT * FROM Filme WHERE taxa <= 10;
```

- LIKE – Utilizado para comparar texto

```
SELECT * FROM ator WHERE nome like 'Emilia';
```

Um nome igual a 'Emilia'

```
SELECT * FROM ator WHERE nome like 'Emi%';
```

Um nome iniciado por 'Emi'

```
SELECT * FROM ator WHERE nome like 'E%';
```

Um nome iniciado pela letra 'E'

```
SELECT * FROM ator WHERE nome like '%a%';
```

Um nome que contenha a letra 'a'

EXPRESSÃO TABLE

WHERE

- BETWEEN – Entre um valor E outro

```
SELECT * FROM Filme WHERE taxa BETWEEN 5 AND 10;
```

- IN – Define uma lista de itens

```
SELECT * FROM Idioma WHERE nome IN ('Portugues','Inglês');
```

```
SELECT * FROM Filme WHERE Filme_id IN (1,2);
```

```
SELECT * FROM ator WHERE ator_id IN  
(SELECT ator_id FROM ator_filme WHERE filme_id = 1);
```

EXPRESSÃO TABLE

GROUP BY

- Após o filtro WHERE, seu conjunto de dados pode ainda precisar de um agrupamento para linhas que tenham os mesmos valores em uma ou mais colunas. Isso é possível utilizando o **GROUP BY** (e o HAVING).
- As redundâncias são eliminadas;
- Permite a utilização de funções de agregação como “soma”, “contagem”, “média”, ...;

```
SELECT * FROM ator_filme;
```

```
SELECT filme_id, COUNT(ator_id) FROM ator_filme GROUP BY filme_id;
```

	filme_id integer	count bigint
1	2	1
2	1	51

	filme_id integer	ator_id integer	ultima_atu timestamp
1	2	54	
2	1	54	
3	1	55	
4	1	56	
5	1	57	
6	1	58	
7	1	59	
8	1	60	
9	1	61	
10	1	62	
11	1	63	
12	1	64	
13	1	65	
14	1	66	
15	1	67	

EXPRESSÃO TABLE

AGGREGATE FUNCTIONS (FUNÇÕES DE AGREGAÇÃO)

○ SQL aggregate functions retornam um valor único, calculado dos valores de uma coluna.

○ Aggregate functions úteis:

- AVG() – Retorna o valor da media;
- COUNT() – Retorna o número de linhas;
- MAX() – Retorna o maior valor;
- MIN() – Retorna o menor valor;
- SUM() – Retorna a soma dos valores;

```
○ SELECT SUM(taxa_aluguel) FROM filme;
```

```
○ SELECT AVG(taxa_aluguel) FROM filme;
```

```
○ SELECT MAX(taxa_aluguel) FROM filme;
```

```
○ SELECT MIN(taxa_aluguel) FROM filme;
```

```
SELECT filme_id, COUNT(ator_id) FROM ator_filme GROUP BY filme_id;
```

EXPRESSÃO TABLE

HAVING

- A cláusula HAVING foi adicionada pois a WHERE não pode ser utilizada com funções de agregação.

```
SELECT filme_id, COUNT(ator_id)
FROM ator_filme
GROUP BY filme_id
HAVING COUNT(ator_id) > 1;
```

	filme_id integer	count bigint
1	1	51

```
SELECT filme_id, COUNT(ator_id)
FROM ator_filme
GROUP BY filme_id;
```

	filme_id integer	count bigint
1	2	1
2	1	51

EXPRESSÃO TABLE

LIMIT E OFFSET

- LIMIT e OFFSET permitem recuperar apenas um subconjunto das linhas que forem geradas pela query;
- A cláusula LIMIT restringe a quantidade máxima de linhas que será recuperada do select:

```
SELECT * FROM ator_filme LIMIT 10;
```

- A cláusula OFFSET diz para pular uma quantidade de linhas antes de retornar as linhas da query.

```
SELECT * FROM ator_filme OFFSET 10;
```

DÚVIDAS?



EXERCÍCIO

1. Escreva um comando SQL que liste os idiomas existentes no banco de dados LocadoraDVD;
 - Resultado Esperado: Português; Inglês; Espanhol; Italiano; Francês; Japonês; Chinês;
2. Escreva um comando SQL que liste os atores existentes no banco de dados LocadoraDVD, em ordem alfabética decrescente;
3. Escreva um comando SQL que mostre a quantidade de cidades por país existentes no banco de dados LocadoraDVD;
4. Escreva um comando SQL que mostre os dados da cidade “Mossoró”. A consulta deve ser formulada pelo campo cidade.
5. Escreva um comando SQL que mostre apenas nome e sobrenome do seu ator favorito. A consulta deve ser formulada a partir das 3 primeiras letras do nome.
6. Escreva um comando SQL que liste a média de taxa de aluguel de todos os seus filmes;
7. Escreva um comando SQL que mostre a quantidade de cidades por país existentes no banco de dados LocadoraDVD;

REFERÊNCIAS BIBLIOGRÁFICAS

PostgreSQL 9.0.22 Documentation. Disponível em:

<<https://www.postgresql.org/files/documentation/pdf/9.0/postgresql-9.0-US.pdf>>. Acesso em 27 Set. 2016.

O Comando Copy. Disponível em:

<http://postgresqlbr.blogspot.com.br/2007/07/o-comando-copy.html>.

Acesso em 15 Nov. 2016.