



Banco de Dados SQL

Bloco 20 - Aula 20.2



Roadmap da Aula



- Select;
- Consulta Select com AS;
- Consulta Select com Concat;
- Consulta Select com DISTINCT;
- Consulta Select com COUNT;
- Consulta Select com LIMIT;
- Consulta Select com OFFSET;
- Consulta Select com ORDER BY;
- Atividade prática;



AS e CONCAT



- **Busque por todos os primeiros e últimos nomes da tabela ator e dê o apelido de "Nome Completo"**
- **Busque por todos os filmes e mostre os dados de nome (NOME DO FILME), duração (DURAÇÃO EM MINUTOS) e classificação (CLASSIFICAÇÃO DO FILME)**





- **Busque por todos os primeiros e últimos nomes da tabela ator e dê o apelido de "Nome Completo"**
 - `SELECT CONCAT(first_name, ' ', last_name) AS 'Nome Completo' FROM sakila.actor;`
- **Busque por todos os filmes e mostre os dados de nome (NOME DO FILME), duração (DURAÇÃO EM MINUTOS) e classificação (CLASSIFICAÇÃO DO FILME)**
 - `USE sakila;`
 - `SELECT title AS 'NOME DO FILME', rental_rate AS 'DURAÇÃO EM MINUTOS', length AS 'CLASSIFICAÇÃO DO FILME' FROM film;`

AS e CONCAT



- Busque todos primeiros e últimos nomes e email das pessoas na tabela customer e monte o seguinte retorno com o nome de Pessoa: Nome Completo: Felipe Nunes | Email: teste@teste.com



AS e CONCAT



- **Busque todos primeiros e últimos nomes e email das pessoas na tabela customer e monte o seguinte retorno com o nome de Pessoa: Nome Completo: Felipe Nunes | Email: teste@teste.com**
 - `SELECT CONCAT ('Nome Completo: ', first_name, ' ', last_name, ' | E-mail: ', email) AS Pessoa FROM sakila.customer;`



DISTINCT



- **Busque por todos os primeiros nomes da tabela ator**
- **Busque por todos os primeiros nomes da tabela ator sem haver nomes repetidos**
- **Busque todos os distritos da tabela address sem haver repetição**
- **Busque todos endereços e distritos da tabela address sem haver repetição**



DISTINCT



- **Busque por todos os primeiros nomes da tabela ator**
 - `SELECT first_name FROM sakila.actor`
- **Busque por todos os primeiros nomes da tabela ator sem haver nomes repetidos**
 - `SELECT DISTINCT first_name FROM sakila.actor;`
- **Busque todos os distritos da tabela address sem haver repetição**
 - `SELECT DISTINCT district FROM sakila.address;`
- **Busque todos endereços e distritos da tabela address sem haver repetição**
 - `SELECT DISTINCT district FROM sakila.address;`



COUNT



- Retorne o total de filmes na tabela film
- Retorne o total de códigos postal da tabela address
- Retorne o total de address_id da tabela address
- Retorne o total de address2 da tabela address



COUNT



- **Retorne o total de filmes na tabela film**
 - `SELECT COUNT(*) FROM sakila.film;`
- **Retorne o total de códigos postal da tabela address**
 - `SELECT COUNT(postal_code) FROM sakila.address;`
- **Retorne o total de address_id da tabela address**
 - `SELECT COUNT(address_id) FROM sakila.address;`
- **Retorne o total de address2 da tabela address**
 - `SELECT COUNT(address2) FROM sakila.address;`



LIMIT e OFFSET



- Retorne os dados da tabela `city` a partir da cidade de `Alessandria` e limitado em 100 registros no máximo
- Retorne os 25 primeiros filmes da tabela `film`



LIMIT e OFFSET



- **Retorne os dados da tabela city a partir da cidade de Alessandria e limitado em 100 registros no máximo**
 - `SELECT * FROM sakila.city LIMIT 100 OFFSET 16;`
- **Retorne os 25 primeiros filmes da tabela film**
 - `SELECT * FROM sakila.film LIMIT 25;`



ORDER BY



- **Retorne o primeiro e último nome da tabela actor em ordem crescente do primeiro nome**
- **Retorne o primeiro e último nome da tabela actor em ordem decrescente do último nome**
- **Retorne o primeiro e último nome da tabela actor em ordem crescente do primeiro nome e decrescente do último nome**



ORDER BY



- **Retorne o primeiro e último nome da tabela actor em ordem crescente do primeiro nome**
 - `SELECT first_name, last_name FROM sakila.actor ORDER BY first_name;`
- **Retorne o primeiro e último nome da tabela actor em ordem decrescente do último nome**
 - `SELECT first_name, last_name FROM sakila.actor ORDER BY last_name DESC;`
- **Retorne o primeiro e último nome da tabela actor em ordem crescente do primeiro nome e decrescente do último nome**
 - `SELECT first_name, last_name FROM sakila.actor ORDER BY first_name ASC, last_name DESC;`



Atividade de Fixação



**Vamos exercitar de forma
conjunta todos
comandos?**



Atividade de Fixação



- **Retorne o primeiro nome da tabela actor como Nome da Pessoa em ordem decrescente do primeiro nome e sem repetições**



Atividade de Fixação



- **Quantos sobrenomes únicos temos na tabela actor?**



Atividade de Fixação



- **Selecione 50 cidades a partir de Angra dos Reis na tabela city?**



Atividade de Fixação



- **Selecione 25 filmes a partir de APOLLO TEEN na tabela film em ordem decrescente de duração do filme (length)**



Atividade de Fixação



- **Selecione 30 atores com seu primeiro e último nome como NOME do ATOR a partir de ZERO CAGE na tabela actor em ordem decrescente do último nome**



Atividade de Fixação



- **Retorne o primeiro nome da tabela actor como Nome da Pessoa em ordem decrescente do primeiro nome e sem repetições**
 - `SELECT DISTINCT first_name AS 'NOME DA PESSOA' FROM sakila.actor ORDER BY first_name DESC;`



Atividade de Fixação



- **Quantos sobrenomes únicos temos na tabela actor?**
 - `SELECT COUNT(DISTINCT last_name) FROM sakila.actor;`



Atividade de Fixação



- **Selecione 50 cidades a partir de Angra dos Reis na tabela city?**
 - `SELECT * FROM sakila.city LIMIT 50 OFFSET 24;`



Atividade de Fixação



- **Selecione 25 filmes a partir de APOLLO TEEN na tabela film em ordem decrescente de duração do filme (length)**
 - `SELECT * FROM sakila.film ORDER BY length DESC LIMIT 25 OFFSET 32;`



Atividade de Fixação



- **Selecione 30 atores com seu primeiro e último nome como NOME do ATOR a partir de ZERO CAGE na tabela actor em ordem decrescente do último nome**
 - `SELECT CONCAT(first_name, ' ', last_name) AS 'NOME DO ATOR' FROM sakila.actor ORDER BY last_name DESC LIMIT 30 OFFSET 10;`





- **Crie um banco de dados denominado trybe**
 - Crie uma tabela denominada `pessoa_estudante` com os seguintes campos: `id` (PK, Auto Incremento), `nome` (varchar e not null), `data de nascimento` (date e not null), `situacao` (tinyint e not null), `descricao` (varchar), `data de criacao` (timestamp com valor default) e `data de atualizacao` (timestamp com valor default)
 - Crie uma tabela denominada `endereco` com os seguintes campos: `id` (PK, Auto Incremento), `endereco` (texto e not null), `cidade` (varchar e not null), `estado` (varchar e not null), `id_pessoa` (chave estrangeira)



Atividade Prática - Extra



```
CREATE DATABASE trybe
```

```
CREATE TABLE IF NOT EXISTS pessoa_estudante (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(150) NOT NULL,  
  data_nascimento DATE,  
  situacao TINYINT NOT NULL,  
  descricao TEXT,  
  data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  data_atualizacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
) ENGINE=INNODB;
```



Atividade Prática - Extra



```
CREATE TABLE IF NOT EXISTS endereco (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    endereco TEXT NOT NULL,  
    cidade VARCHAR(255) NOT NULL,  
    estado VARCHAR(255) NOT NULL,  
    id_pessoa INT,  
    FOREIGN KEY (id_pessoa)  
        REFERENCES pessoa_estudante (id)  
    ON UPDATE RESTRICT ON DELETE CASCADE  
);
```

