



**UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS JARDINS DE ANITA
TÉCNOLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

JÔNATAS FERNANDES SILVA

**TRABALHO 02
MYSQL**

**ITAPAJÉ
2023**

Todos os códigos exibidos a seguir podem ser encontrados no repositório abaixo
Github -> <https://github.com/jonatasfernandessilva7/atividades-banco-de-dados-mysql.git>

BANCO LOCADORA

#

#esse é pra valer..

– criando banco de dados

CREATE DATABASE locadora;

USE locadora;

– criando tabelas

**CREATE TABLE clientes(
cli_codigo INT NOT NULL,
cli_nome VARCHAR(110) NOT NULL,
cli_cpf CHAR(12),
cli_data_nasc DATE, #atributo do tipo data
cli_sexo CHAR(1),
cli_email VARCHAR(110)
);**

**CREATE TABLE filmes(
film_codigo INT NOT NULL,
film_titulo VARCHAR(110) NOT NULL,
film_genero VARCHAR(50),
film_duracao TIME, #atributo de tempo
film_status VARCHAR(12),
film_preco NUMERIC(3,2) #indica que pode ter até 3 caracteres sendo 2 deles
depois da vírgula pode ter até 2 caracteres
);**

– apagando tabela filmes

drop table filmes;

#

#descrevendo as tabelas existentes

DESC clientes;

	Field	Type	Null	Key	Default	Extra
►	cli_codigo	int	NO		NULL	
	cli_nome	varchar(110)	NO		NULL	
	cli_cpf	char(12)	YES		NULL	
	cli_data_nasc	date	YES		NULL	
	cli_sexo	char(1)	YES		NULL	
	cli_endereco_eletronico	varchar(80)	YES		NULL	
	cli_endereco	varchar(100)	YES		NULL	

DESC filmes;

	Field	Type	Null	Key	Default	Extra
►	film_codigo	int	NO		NULL	
	film_titulo	varchar(110)	NO		NULL	
	film_genero	varchar(50)	YES		NULL	
	film_duracao	time	YES		NULL	
	film_status	varchar(12)	YES		NULL	
	film_preco	decimal(3,2)	YES		NULL	

– inserindo dados nas tabelas

INSERT INTO clientes

VALUES(112, 'antonio robreto amadeu', '112112112-12', '1250-02-12', 'f', 'antoniocavaloamadeuroberto@gmail.oprkut.gb');

INSERT INTO filmes

VALUES(3,'gta sa', 'reallife', '16:58:59', 'on', 8.5);

INSERT INTO clientes

VALUES(112, 'antonio bgxdvasghdch amadeu', '656575656-12', '1250-02-12', 'm', 'bg\hdfchg\sd@gmail.oprkut.gb');

INSERT INTO filmes

VALUES(5,'e o vento levou', 'eita que filme', '16:58:59', 'on', 8.5);

– buscando dados nas tabelas

SELECT cli_nome, cli_email FROM clientes WHERE cli_sexo = 'm';

SELECT film_codigo , film_titulo FROM filmes;

SELECT film_duracao FROM filmes WHERE film_titulo='e o vento levou';

SELECT film_titulo FROM filmes WHERE film_preco < 3;

SELECT film_titulo FROM filmes WHERE film_status <> 'alugado';

**SELECT cli_nome
FROM clientes WHERE cli_data_nasc >= '1990-1-1';**

**SELECT film_titulo
FROM filmes
WHERE film_preco < 3 AND film_status <> 'alugado';**

**SELECT film_titulo
FROM filmes WHERE film_genero= 'romance' OR film_genero
= 'animação';**

**SELECT film_titulo
FROM filmes WHERE film_codigo BETWEEN 2 AND 20;**

SELECT * FROM clientes WHERE cli_codigo=112;

SELECT * FROM clientes;

SELECT cli_nome, cli_cpf FROM clientes;

SELECT * FROM filmes;

**SELECT cli_email
FROM clientes WHERE cli_nome BETWEEN 'A' AND 'G';**

**SELECT cli_nome
FROM clientes WHERE cli_email LIKE '%gmail.com';**

SELECT * FROM clientes WHERE cli_nome LIKE 'Luiz%';

**SELECT film_titulo
FROM filmes WHERE film_genero IN ('comedia', 'romance',
'acao');**

**SELECT cli_nome
FROM clientes WHERE cli_email IS NULL;**

– resultado final da tabela clientes

cli_codigo	cli_nome	cli_cpf	cli_data_nasc	cli_sexo	cli_endereco_eletronico	cli_endereco
112	antonio robreto amadeu	112112112-12	1250-02-12	f	antoniocavaloamadeuroberto@gmail.oprkut.gb	NULL
112	antonio bgxdvasghdch amadeu	656575656-12	1250-02-12	m	bgghdfchgsd@gmail.oprkut.gb	NULL

– resultado final tabela filmes

nim_codigo	nim_titulo	nim_genero	nim_duracao	nim_status	nim_preco
3	gta sa	reallife	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50
5	e o vento levou	eita que filme	16:58:59	on	8.50

– atualizando dados nas tabelas

UPDATE clientes SET cli_nome = 'jose gabriela bezerra' WHERE cli_nome = 'antonio robreto amadeu';

– deletando dados nas tabelas

DELETE FROM clientes WHERE cli_codigo=112;

#

#reescrevendo queries

SELECT film_titulo FROM filmes WHERE film_preco > 2.5;

SELECT cli_nome FROM clientes WHERE cli_sexo = 'm';

SELECT film_titulo FROM filmes WHERE film_genero IN ('comedia', 'policial');

SELECT cli_nome FROM clientes WHERE cli_nome LIKE 'A%' AND cli_nome LIKE 'B%';

consultas e ordenação de dados

```
SELECT cli_nome, cli_email FROM clientes ORDER BY cli_nome;
```

```
SELECT cli_nome, cli_email FROM clientes ORDER BY cli_nome DESC;
```

```
SELECT cli_nome, cli_email FROM clientes WHERE cli_nome LIKE 'a%'
ORDER BY cli_nome;
```

exercícios

-- 1) exibir nome e CPF de todos os clientes em ordem alfabética

```
SELECT cli_nome FROM clientes ORDER BY cli_nome;
```

-- 2) exibir todas as informações sobre os filmes da locadora em ordem decrescente do código do filme

```
SELECT * FROM filmes ORDER BY film_codigo DESC;
```

-- 3) exibir o nome e email dos clientes do sexo masculino em ordem alfabética inversa

```
SELECT cli_nome, cli_email FROM clientes WHERE cli_sexo = 'm' ORDER
BY cli_nome DESC;
```

-- 4) exibir o nome e a data de nascimento dos clientes que usam email do hotmail (@hotmail.com) em ordem alfabética

```
SELECT cli_nome, cli_data_nasc FROM clientes WHERE cli_email LIKE
'%hotmail.com' ORDER BY cli_nome;
```

#

consultas usando agrupamento de dados

```
SELECT film_preco FROM filmes GROUP BY film_preco;
```

	film_preco
▶	8.50

```
SELECT COUNT(film_titulo) FROM filmes;
```

	COUNT(film_titulo)
▶	9

SELECT COUNT(DISTINCT film_titulo) FROM filmes;

	COUNT(DISTINCT film_titulo)
▶	2

SELECT MIN(film_preco), AVG(film_preco), MAX(film_preco) FROM filmes;

	MIN(film_preco)	AVG(film_preco)	MAX(film_preco)
▶	8.50	8.500000	8.50

exercícios

-- 1) quantos clientes do sexo feminino têm na locadora

SELECT cli_sexo, COUNT(cli_sexo) FROM clientes GROUP BY cli_sexo;

	cli_sexo	COUNT(cli_sexo)
▶	f	1
	m	1

-- 2) quantos exemplares de um mesmo filme estão cadastrados na locadora

SELECT film_titulo, COUNT(film_titulo) AS Unidades FROM filmes GROUP BY film_titulo;

	film_titulo	Unidades
▶	gta sa	1
	e o vento levou	8

SELECT film_titulo AS titulo, COUNT(film_titulo) AS quantidade FROM filmes GROUP BY film_titulo HAVING COUNT(film_titulo)>1;

	titulo	quantidade
▶	e o vento levou	8

exercicios parte 02

```
-- * -- * -- * -- * -- * -- * --
```

SELECT film_status, COUNT(film_status) FROM filmes GROUP BY film_status;

	film_status	COUNT(film_status)
▶	on	9

SELECT film_genero AS genero, COUNT(film_genero) AS quantidade FROM filmes GROUP BY film_genero;

	genero	quantidade
▶	reallife	1
	eita que filme	8

SELECT film_preco AS preco_unitario, SUM(film_preco) AS total FROM filmes GROUP BY film_preco;

	preco_unitario	total
▶	8.50	76.50

```
-- * -- * -- * -- * -- * -- * --
```

-- comando alter

ALTER TABLE clientes ADD cli_endereco VARCHAR(100);

**ALTER TABLE filmes
ADD fil_direcao VARCHAR(50)**

FIRST;

**ALTER TABLE filmes
ADD fil_atores VARCHAR(200)
AFTER fil_direcao;**

**ALTER TABLE filmes DROP COLUMN fil_direcao;
ALTER TABLE filmes DROP COLUMN fil_atores;**

**ALTER TABLE clientes CHANGE COLUMN cli_email
cli_endereco_eletronico varchar(80);**

#

BANCO CINEONLINE

-- criando database

CREATE DATABASE cineOnline;

USE cineOnline;

-- criando tabelas

**CREATE TABLE cinema
(
cinema_codigo int NOT NULL,
cinema_nome varchar(40) NOT NULL,
cinema_sala varchar(2) NOT NULL,
cinema_capacidade int NOT NULL,
cinema_cidade varchar(50) NOT NULL,
PRIMARY KEY(cinema_codigo)
);**

**CREATE TABLE filmes
(
fil_codigo int NOT NULL,
fil_titulo varchar(50) NOT NULL,
fil_genero varchar(30) NOT NULL,
fil_censura char(8) NOT NULL DEFAULT 'Livre',
PRIMARY KEY(fil_codigo)
);**

```
CREATE TABLE sessao
(
ses_codigo int NOT NULL,
ses_codfilme int NOT NULL,
ses_codcinema int NOT NULL,
ses_datahora datetime NOT NULL,
ses_preco numeric(9,2) NOT NULL,
PRIMARY KEY (ses_codigo),
FOREIGN KEY (ses_codfilme) REFERENCES filmes (fil_codigo),
FOREIGN KEY (ses_codcinema) REFERENCES cinema (cinema_codigo)
);
```

```
CREATE TABLE clientes
(
cli_codigo int NOT NULL,
cli_nome varchar(50) NOT NULL,
cli_email varchar (60) NOT NULL,
cli_compras int NOT NULL,
PRIMARY KEY (cli_codigo),
FOREIGN KEY (cli_compras) REFERENCES compras (comp_cod)
);
```

```
CREATE TABLE compras
(
comp_cod int NOT NULL,
comp_valor int NOT NULL,
comp_nome varchar(50) NOT NULL,
comp_fil int NOT NULL,
PRIMARY KEY (comp_cod),
FOREIGN KEY (comp_fil) REFERENCES filmes (fil_codigo)
);
```

-- inserindo dados

```
INSERT INTO cinema (cinema_codigo,cinema_nome, cinema_sala,
cinema_capacidade, cinema_cidade) VALUES(1, 'brasao', '02', 8, 'iataboanacacaca');
INSERT INTO cinema (cinema_codigo,cinema_nome, cinema_sala,
cinema_capacidade, cinema_cidade) VALUES(2, 'iuy', '03', 15, 'iataboanacacaca');
```

```
INSERT INTO filmes (fil_codigo,fil_titulo, fil_genero, fil_censura) VALUES(1, 'vish que bom', 'adolescente', 18);
```

```
INSERT INTO filmes (fil_codigo,fil_titulo, fil_genero, fil_censura) VALUES(2, 'vish que ruim', 'jovem', 29);
```

```
INSERT INTO sessao (ses_codigo,ses_codfilme, ses_codcinema,
ses_datahora,ses_preco) VALUES(1, 2, 1, '1888-09-10', 19.85);
INSERT INTO sessao (ses_codigo,ses_codfilme, ses_codcinema,
ses_datahora,ses_preco) VALUES(2, 2, 1, '1898-10-09', 19.90);
```

```
INSERT INTO compras (comp_cod,comp_valor, comp_nome, comp_fil) VALUES(1, 20,
'vish que bom', 1);
INSERT INTO compras (comp_cod,comp_valor, comp_nome, comp_fil) VALUES(2, 22,
'vish que ruim', 2);
```

```
INSERT INTO clientes (cli_codigo,cli_nome, cli_email, cli_compras) VALUES(1, 'joao
barros', 'joaobarros@hotmail.com',1);
INSERT INTO clientes (cli_codigo,cli_nome, cli_email, cli_compras) VALUES(2, 'joao
silva', 'joaosilva@hotmail.com',2);
```

-- buscando os dados

```
SELECT * FROM cinema;
```

Result Grid					
Filter Rows:					
Edit: Export/Import					
	cinema_codigo	cinema_nome	cinema_sala	cinema_capacidade	cinema_cidade
▶	1	brasao	02	8	iataboanacacaca
	2	iuy	03	15	iataboanacacaca
✱	NULL	NULL	NULL	NULL	NULL

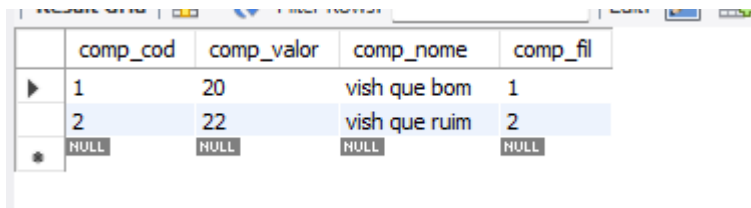
```
SELECT * FROM filmes;
```

	fil_codigo	fil_titulo	fil_genero	fil_censura
▶	1	vish que bom	adolescente	18
	2	vish que ruim	jovem	29
✱	NULL	NULL	NULL	NULL

```
SELECT * FROM sessao;
```

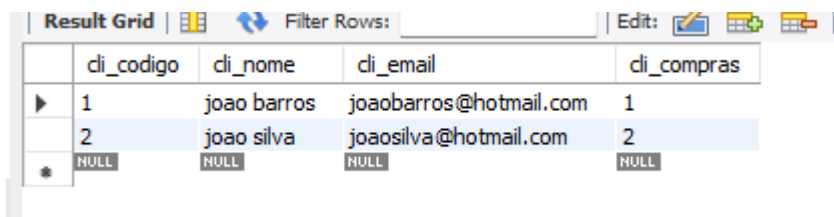
	ses_codigo	ses_codfilme	ses_codcinema	ses_datahora	ses_preco
▶	1	2	1	1888-09-10 00:00:00	19.85
	2	2	1	1898-10-09 00:00:00	19.90
✱	NULL	NULL	NULL	NULL	NULL

SELECT * FROM compras;



	comp_cod	comp_valor	comp_nome	comp_fil
▶	1	20	vish que bom	1
	2	22	vish que ruim	2
*	NULL	NULL	NULL	NULL

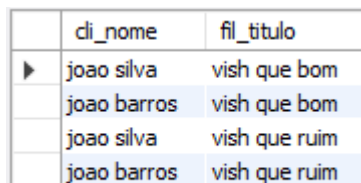
SELECT * FROM clientes;



	cli_codigo	cli_nome	cli_email	cli_compras
▶	1	joao barros	joaobarros@hotmail.com	1
	2	joao silva	joaosilva@hotmail.com	2
*	NULL	NULL	NULL	NULL

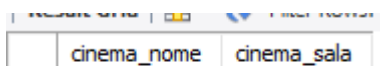
-- JUNTANDO TABELAS

**SELECT cli_nome, fil_titulo
FROM clientes CROSS JOIN filmes;**



	cli_nome	fil_titulo
▶	joao silva	vish que bom
	joao barros	vish que bom
	joao silva	vish que ruim
	joao barros	vish que ruim

**SELECT cinema_nome, cinema_sala
FROM filmes, sessao, cinema
WHERE fil_titulo = 'Procurando Nemo'
AND ses_codfilme = fil_codigo
AND cinema_codigo = ses_codcinema;**



	cinema_nome	cinema_sala
--	-------------	-------------

**SELECT fil_titulo, cinema_nome, cinema_sala
FROM filmes, sessao, cinema
WHERE fil_titulo <> 'Procurando Nemo'
AND ses_codfilme = fil_codigo**

AND cinema_codigo = ses_codcinema;

	fil_titulo	cinema_nome	cinema_sala
▶	vish que ruim	brasao	02
	vish que ruim	brasao	02

SELECT * FROM clientes CROSS JOIN filmes;

	cli_codigo	cli_nome	cli_email	cli_compras	fil_codigo	fil_titulo	fil_genero	fil_censura
▶	2	joao silva	joaosilva@hotmail.com	2	1	vish que bom	adolescente	18
	1	joao barros	joaobarros@hotmail.com	1	1	vish que bom	adolescente	18
	2	joao silva	joaosilva@hotmail.com	2	2	vish que ruim	jovem	29
	1	joao barros	joaobarros@hotmail.com	1	2	vish que ruim	jovem	29

SELECT cinema_nome, cinema_sala, fil_titulo FROM cinema, filmes WHERE fil_titulo = 'vish que bom';

	cinema_nome	cinema_sala	fil_titulo
▶	brasao	02	vish que bom
	iuy	03	vish que bom

----- exercício -----

-- Qual a diferença entre o comando SELECT com conexão cartesiana e o comando SELECT simples que utilizamos nas aulas anteriores?

-- R: a diferença é que o comando select simples busca apenas informações de uma tabela, enquanto o select com conexão cartesiana busca em duas ou mais tabelas de uma só vez

-- Qual o resultado da seguinte consulta: SELECT * FROM clientes CROSS JOIN filmes?

-- R: <https://github.com/jonatasfernandessilva7/atividades-banco-de-dados-mysql.git> (arquivo com nome: resultado select cross join.csv)

-- Qual o resultado da seguinte consulta: SELECT cinema_nome, cinema_sala, fil_titulo FROM cinema, filmes WHERE fil_titulo = 'Procurando Nemo' GROUP BY cinema_nome?

-- R: <https://github.com/jonatasfernandessilva7/atividades-banco-de-dados-mysql.git> (arquivo com nome: resultado select group by.csv)

-- Compare o resultado obtido na consulta realizada na Questão 3 com a consulta do Exemplo 1.

-- R: no exemplo 1, temos como resultado obtido apenas o nome do cinema e a sala onde está sendo exibido o filme

-- R: Já na consulta da questão da questão 3, são obtidos somente o nome e as salas do cinema em que são exibidos o filme da condição where, além de exibir também o nome do filme

-- Você pode utilizar funções como SUM e AVG nas consultas com conexão? Cite exemplos.

-- R: sim.

-- exemplos

-- > SELECT comp_fil AS filme, MAX(comp_fil) AS maior_cod_filme FROM compras, filmes, sessao, cinema WHERE fil_titulo <> 'procurando nemo' AND ses_codfilme = fil_codigo AND cinema_codigo = ses_codcinema GROUP BY comp_fil;