

Script SQL

```
SET SQL_SAFE_UPDATES = 0;
```

```
SET SQL_SAFE_UPDATES = 1;
```

```
/*Comando "CREATE DATABASE" permite criar um banco de dados*/  
CREATE DATABASE db_cadastro;
```

```
/*Comando "USE" permite selecionar o banco de dados para utilizar*/  
USE db_cadastro;
```

```
/*Comando "CREATE TABLE" permite criar uma tabela no banco de dados*/  
CREATE TABLE tb_pessoas(  
id int not null auto_increment,  
nome varchar(30) NOT NULL,  
nascimento date,  
sexo enum('M','F'),  
peso decimal(5,2),  
altura decimal(3,2),  
nacionalidade varchar(20) default 'Brasil',  
primary key(id)  
);
```

```
/*Comando "DESCRIBE" permite ver a estrutura interna de uma tabela*/  
DESCRIBE tb_pessoas;
```

```
DESC tb_pessoas;
```

```
/*Comando "DROP DATABASE" tem a função de apagar o banco de dados*/  
DROP DATABASE db_cadastro;
```

```
/*Comando "DROP TABLE" tem a função de apagar uma tabela que está no banco de dados*/  
DROP TABLE tb_pessoas;  
DROP TABLE tb_cursos;
```

```
/*Comando "INSERT INTO" permite inserir registros na tabela*/  
INSERT INTO tb_pessoas (id, nome, nascimento, sexo, peso, altura, nacionalidade)  
VALUES  
( '1','Godofredo', '1984-01-02', 'M', '78.5', '1.83', 'Brasil');
```

/*Como vou inserir os dados nas ordens que os campos foram criados na tabela, não preciso colocar os CAMPOS no comando "INSERT INTO*/

```
INSERT INTO tb_pessoas  
VALUES  
( default,'Neuza', '1964-06-20', 'F', '57.6', '1.60', 'Brasil');
```

/*Comando "SELECT * FROM " serve para realizar consultas nas tabelas*/

```
SELECT * FROM tb_pessoas;
```

/*Inserindo varios dados na tabela ao mesmo tempo*/

```
INSERT INTO tb_pessoas (nome, nascimento, sexo, peso, altura, nacionalidade)  
VALUES  
( 'Ana', '1975-12-22', 'F', '52.3', '1.45', 'EUA'),  
( 'Pedro', '2000-07-15', 'M', '52.3', '1.45', 'Brasil'),  
( 'Maria', '1999-05-30', 'F', '75.9', '1.70', 'Portugal');
```

/*Comando "ALTER TABLE" permite alterar a estrutura da tabela

Comando "ADD COLUMN" permite adicionar um novo campo na tabela*/

```
ALTER TABLE tb_pessoas  
ADD COLUMN profissao VARCHAR(10); /*Adicionei um campo chamado "profissao" na  
tabela "pessoas"*/
```

/*Comando "ALTER TABLE" permite alterar a estrutura da tabela

Comando "DROP TABLE" permite eliminar uma coluna/campo da tabela no MySQL*/

```
ALTER TABLE tb_pessoas  
DROP COLUMN profissao;
```

/*Comando "ALTER TABLE" permite alterar a estrutura da tabela

Comando "ADD COLUMN" permite adicionar um novo campo na tabela

Comando "AFTER" permite escolher uma posição da coluna/campo na tabela*/

```
ALTER TABLE tb_pessoas  
ADD COLUMN profissao VARCHAR(10) AFTER nome; /*"AFTER" indica que depois do  
campo nome será inserido o novo campo profissão*/
```

/*O comando "FIRST" permite eu adicionar um campo novo no primeiro campo da tabela

Ex: O primeiro campo da tabela era "id" depois de utilizar o "FIRST" o primeiro campo tornou-se o campo "codigo"*/

```
ALTER TABLE tb_pessoas  
ADD COLUMN codigo INT FIRST; /*Criei o campo "codigo" e deixei ele sendo o primeiro  
campo/coluna da tabela pessoas*/
```

/*Alterando o tipo do campo, nesse caso o campo profissao e do tipo varchar(10) e estou alterando para varchar(20)*/

```
ALTER TABLE tb_pessoas  
MODIFY COLUMN profissao VARCHAR(20);
```

/*Renomeando a coluna/campo chamado "profissao" para "prof"*/

```
ALTER TABLE tb_pessoas  
Change column profissao prof VARCHAR(20);
```

/*Renomeando a tabela "tb_pessoas" para "tb_gafanhotos"*/

```
ALTER TABLE tb_pessoas  
RENAME TO tb_gafanhotos;
```

/*Criando uma tabela utilizando "IF NOT EXISTS" ou "IF EXISTS"

IF NOT EXISTS permite criar uma tabela caso ela não exista

IF EXISTS permite criar uma tabela caso ela exista*/

```
CREATE TABLE IF NOT EXISTS tb_cursos(  
nome VARCHAR(30) NOT NULL UNIQUE,  
descricao TEXT,  
carga INT UNSIGNED,  
totaulas INT,  
ano YEAR DEFAULT '2016'  
);
```

/*Adicionando uma chave primaria para uma tabela já existente.Vou adicionar uma chave primaria na tabela "tb_cursos" que já foi criada anteriormente*/

/*1° Passo: Colocar o campo "idcurso " como sendo o primeiro campo da tabela*/

```
ALTER TABLE tb_cursos  
ADD COLUMN idcurso INT FIRST;
```

```
SELECT * FROM tb_cursos;
```

```
DESCRIBE tb_cursos;
```

/*2° Passo: Colocar o campo/coluna "idcurso" como chave primaria*/

```
ALTER TABLE tb_cursos  
ADD PRIMARY KEY(idcurso);
```

/*Inserindo dados na tabela "tb_cursos"*/

```
INSERT INTO tb_cursos (idcurso,nome, carga, ano)  
VALUES  
(1,'HTML4',40,'2014');
```

```
INSERT INTO tb_cursos (nome, carga, ano)
VALUES
('HTML4','40','2014');
```

```
insert into tb_cursos values
('1', 'HTML4', 'Curso de HTML5', '40', '37', '2014'),
('2', 'Algoritmos', 'Lógica de Programação', '20', '15', '2014'),
('3', 'Photoshop', 'Dicas de Photoshop CC', '10', '8', '2014'),
('4', 'PGP', 'Curso de PHP para iniciantes', '40', '20', '2010'),
('5', 'Java', 'Introdução à Linguagem Java', '10', '29', '2000'),
('6', 'MySQL', 'Banco de Dados MySQL', '30', '15', '2016'),
('7', 'Word', 'Curso completo de Word', '40', '30', '2016'),
('8', 'Sapateado', 'Danças Rítmicas', '40', '37', '2018'),
('9', 'Cozinha Árabe', 'Aprender a fazer Kibe', '40', '30', '2018'),
('10', 'YouTuber', 'Gerar polêmica e ganhar inscritos', '5', '2', '2018');
```

/*Excluindo o registro que se encontra no id 1 da tabela tb_cursos, ou seja, excluindo o dado que inserimos do curso*/

```
DELETE FROM tb_cursos WHERE idcurso = 1; /*Excluindo informações do curso cujo o id seja igual a 1*/
```

/*Comando "UPDATE" permite alterar/modificar os dados de um campo de uma tabela*/

```
UPDATE tb_cursos
SET nome = 'HTML5' /*Estou modificando o registro/informação do campo da tabela cursos de "HTML4" para "HTML5"*/
WHERE idcurso = '1';
```

```
UPDATE tb_cursos
SET nome = 'PHP', ano = '2015'
WHERE idcurso = '4';
```

```
UPDATE tb_cursos
SET nome = 'Java', carga = '40', ano = '2015'
WHERE idcurso = '5';
```

/*Parametro "LIMIT" permite limitar o números de linhas que serão afetadas com a modificação/alteração do registro*/

```
UPDATE tb_cursos
SET nome = 'Java', carga = '40', ano = '2015'
WHERE idcurso = '5'
LIMIT 1; /*Nesse caso vai limitar as modificações/alterações a somente uma linha, a linha de id = 5
```

/*Deletando o registro/linha da tabela cursos*/

DELETE FROM tb_cursos

WHERE idcurso = '8'; **/*Deletei o curso da tabela cursos que o id era igual a 8*/**

/*Apagando todos os registros linhas que tem o ano igual a 2018*/

DELETE FROM tb_cursos

WHERE ano = '2018';

/*Apagando apenas um registro (uma linha) que tem o ano igual a 2018*/

DELETE FROM tb_cursos

WHERE ano = '2018'

LIMIT 1;

/*Comando "TRUNCATE" permite excluir todos os registros/dados/linhas presentes na tabela*/

TRUNCATE TABLE tb_cursos;

/*Comando para ver tabelas*/

SHOW TABLES;

/*Comando para ver os bancos de dados*/

SHOW DATABASES;

/*A palavra-chave ORDER BY é usada para ordenar o conjunto-resultado de registro em uma consulta SQL, tanto de forma crescente quanto decrescente*/

SELECT * FROM tb_cursos

ORDER BY nome ASC; **/*Estou ordenando de forma alfabética os nomes dos cursos da tabela cursos*/**

SELECT * FROM tb_cursos

ORDER BY nome DESC; **/*Estou ordenando de forma alfabética inversa os nomes dos cursos da tabela cursos */**

SELECT * FROM tb_cursos

ORDER BY nome; **/*Por padrão sem colocar "ASC" ou "DESC" a consulta é ordenada de forma alfabética os nomes dos cursos da tabela cursos*/**

```
SELECT nome, carga /*Vai mostrar na consulta os campos "nome" e "carga"*/  
FROM tb_cursos  
ORDER BY carga; /*Estou ordenando de forma crescente "ASC" utilizando a carga como  
forma de ordenar, como e número será do menor para o maior*/  
/*Nesse caso não precisa colocar "ASC" devido por padrão  
ele ordena de forma crescente*/
```

```
SELECT nome, carga  
FROM tb_cursos  
ORDER BY carga DESC; /*Estou ordenando de forma decrescente "DESC" utilizando a carga  
como forma de ordenação, ou seja, vai ordenar conforme números de cargas, nesse caso do  
maior para o menor*/
```

```
/*Selecionando/Consultando somente os campos/colunas "nome,carga e ano"*/  
SELECT nome, carga, ano  
FROM tb_cursos  
ORDER BY nome; /*Será ordenado pelo campo "nome" essa consulta, como estou  
ordenando por caractere, ou seja, pelo nome do curso por padrão vai ser ordenado de  
forma alfabética*/
```

```
SELECT carga, nome, ano  
FROM tb_cursos  
ORDER BY carga; /*Aqui estou ordenando pela coluna "carga", como e um número será  
ordenado por padrão do menor para o maior*/
```

```
SELECT carga, nome, ano  
FROM tb_cursos  
ORDER BY ano, nome; /*Primeiro vai ser ordenado pelo campo "ano" e depois pelo campo  
"nome"*/
```

```
/*Selecione todos os campos da tabela cursos onde o ano seja igual a 2016 ordenado  
por nome, como o padrão do "ORDER BY" e "ASC" será ordenado de forma  
alfabética*/
```

```
SELECT * FROM tb_cursos  
WHERE ano = '2016'  
ORDER BY nome;
```

```
/*Estou consultando somente as colunas/campo "nome e carga" da tabela cursos  
onde o ano seja igual a 2016 ordenado por nome*/
```

```
SELECT nome, carga  
FROM tb_cursos  
WHERE ano = '2016'  
ORDER BY nome;
```

```
SELECT nome, descricao
FROM tb_cursos
WHERE ano <= '2015' /*Menor ou igual */
ORDER BY nome;
```

```
SELECT nome, descricao
FROM tb_cursos
WHERE ano < '2015' /*Vai mostrar os cursos que foram lançados antes de 2015*/
ORDER BY nome;
```

```
SELECT nome, descricao
FROM tb_cursos
WHERE ano > '2016' /*Vai mostrar os cursos que foram lançados depois do ano 2016*/
ORDER BY ano, nome; /*Vai ordenar primeiramente por ano, caso os anos sejam iguais
será ordenado por nome*/
```

```
SELECT nome, descricao
FROM tb_cursos
WHERE ano >= '2016' /*Vai mostrar os cursos lançados no ano de 2016 e os anos depois de
2016*/
ORDER BY ano, nome;
```

```
SELECT nome, descricao, ano
FROM tb_cursos
WHERE ano != '2016' /*Vai mostrar os campos (nome, descricao, ano) da tabela cursos que
foram lançados em todos os anos, menos no ano de 2016*/
ORDER BY ano, nome;
```

/*A cláusula BETWEEN permite selecionar intervalos de dados ao retornar os resultados de uma consulta */

```
SELECT * FROM tb_cursos
WHERE totaulas BETWEEN '20' AND '30' /*Quero consultar/mostrar todos os campos da
tabela cursos onde o total de aula seja entre 20 a 30 horas, vai ser ordenado pelo campo
nome de forma alfabetica*/
ORDER BY nome;
```

```
SELECT nome, ano
FROM tb_cursos
WHERE ano
BETWEEN 2014 and 2016; /*Quero que mostre as colunas/campos "nome e ano" dos
cursos que foram lançados no periodo de 2014 a 2016*/
```

/*Quero que mostre na consulta as colunas nome e ano da tabela cursos que foram lançados no periodo de 2014 a 2016 ordenado por ano de forma decrescente "DESC", ou seja, como e número será do maior para o menor, e ordenar por nome de forma crescente "ASC" que e o padrão do ORDER BY, ordenar de forma ASC caracteres/nomes e deixar ordenado de forma alfabética*/

```
SELECT nome, ano
FROM tb_cursos
WHERE ano
BETWEEN 2014 AND 2016
ORDER BY ano DESC, nome;
```

/*Mostre na tela os campos/colunas "idcurso e nome" da tabela cursos do ano 2014,2016 e 2018 somente, não quero que mostre outros anos, somente esses que especifiquei*/

```
SELECT idcurso, nome
FROM tb_cursos
WHERE ano IN ('2014','2016','2018')
ORDER BY nome;
```

/*Mostre na tela os campos/colunas "idcurso e nome" da tabela cursos, cuja o curso NAO foi lançado no ano 2014,2016 e 2018,ou seja, quero que mostre todos os cursos que não foram lançados no ano de "2014","2016" e "2018" */

```
SELECT idcurso, nome
FROM tb_cursos
WHERE ano NOT IN('2014','2016','2018')
ORDER BY nome;
```

```
SELECT nome, descricao, ano
FROM tb_cursos
WHERE ano IN ('2014','2016') /*Vai mostrar os campos "nome, descricao, ano" dos cursos que foram lançados em 2014 e 2016*/
ORDER BY ano;
```

```
SELECT nome, descricao, ano
FROM tb_cursos
WHERE ano NOT IN ('2014','2016') /*Vai mostrar os campos "nome, descricao, ano" dos cursos que foram lançados em todos anos, menos nos anos de 2014 e 2016*/
ORDER BY ano;
```


/*Vai mostrar na tela todos campos da tabela cursos que tenham carga maior que 30 e total de aulas menor que 30 horas*/

```
SELECT * FROM tb_cursos  
WHERE carga > 35 AND totaulas < 30  
ORDER BY nome;
```

```
SELECT nome, carga, totaulas  
FROM tb_cursos  
WHERE carga >= 35 AND totaulas <= 30 /*Vai mostrar os cursos com carga maior ou  
igual a 35 e (AND) totaulas menor ou igual a 30*/  
ORDER BY nome;
```

/*Mostre na tela o nome, a carga e o totaulas da tabela curso onde a carga seja maior que 35 OU o total de aulas menor que 30.

Nesse caso como estamos utilizando o "OR" que significa "ou", os dois não precisam atender os requisitos, um deles já atendendo vai ser mostrado na tela*/

```
SELECT nome, carga, totaulas  
from tb_cursos  
WHERE carga > 35 or totaulas < 30;
```

/*Mostre/selecione todos os dados da tabela cursos onde o nome seja igual a PHP*/

```
SELECT * FROM tb_cursos  
WHERE nome = 'PHP';
```

/*Vai mostrar todos os cursos que o campo "nome" comecam com a letra 'P'*/

```
SELECT * FROM tb_cursos  
WHERE nome LIKE 'P%'; /*Si utilizar "P" ou "p" vai dar no mesmo*/
```

/*Vai mostrar todos os cursos que o campo " nome" NÃO comecam com a letra "P"*/

```
SELECT * FROM tb_cursos  
WHERE nome NOT LIKE 'P%';
```

/*Vai mostrar todos os cursos que tem o campo "nome" terminado com a letra "A"*/

```
SELECT * FROM tb_cursos  
WHERE nome LIKE '%A';
```

/*Vai mostrar os campos "nome e descricao" dos cursos cujo o nome inicie com qualquer letra, e a segunda letra seja "y" e o restante seja qualquer letras*/

```
SELECT nome, descricao  
FROM tb_cursos  
WHERE nome LIKE '_y%';
```

/*Nesse caso ele vai selecionar qualquer registro, nome do curso que tem a letra "a" ou "A" em qualquer lugar.

Os que começam com "A"

Os que terminam com "A"

Os que tem dois "A" no meio

Inclusive aquele "Á" acentuado também e considerado dentro do select */

```
SELECT * FROM tb_cursos
```

```
WHERE nome LIKE '%A%';
```

/*Quero que mostre todos os cursos que o nome do curso não tem "A" ou "a" ou "á" ou "Á" em algum lugar*/

```
SELECT * FROM tb_cursos
```

```
WHERE nome NOT LIKE '%A%';
```

/*Vai mostrar/selecionar qualquer curso cuja o nome inicie com "PH" e termine com "P" ou "p"*/

```
SELECT * FROM tb_cursos
```

```
WHERE nome LIKE 'PH%P';
```

```
INSERT INTO tb_cursos (idcurso, nome, descricao, carga, totaulas, ano)
```

```
VALUES
```

```
(11, 'PHP1', 'Testando o Select','10', 50, 2021),
```

```
(12, 'PhyyypP1', 'Testando o Select','11', 90, 2031),
```

```
(13, 'PHPAPA', 'Testando o Select','13', 70, 2026);
```

```
desc tb_cursos;
```

/*Quero selecionar todos os cursos que o nome inicie com "ph" tem qualquer coisa no meio em seguida termine com "p" e depois do "p" tenha alguma letra*/

```
SELECT * FROM tb_cursos
```

```
WHERE nome like 'ph%p_';
```

/*Selecionar qualquer curso que o nome inicie com a letra "P" tenha qualquer letra depois e tenha outro "P" e qualquer coisa depois desse ultimo "p"*/

```
SELECT * FROM tb_cursos
```

```
WHERE nome like 'p_p%';
```

/*Selecionar qualquer curso que o nome inicie com a letra "P" tenha qualquer letraS depois e tenha outro "P" e qualquer coisa depois desse ultimo "p"*/

```
SELECT * FROM tb_cursos
```

```
WHERE nome like 'p%p%';
```

/*Selecione qualquer curso que o nome inicie com a letra "P" tenha 2 letras qualquer em seguida, e tenha um "t" e qualquer coisa depois*/

```
SELECT * FROM tb_cursos  
WHERE nome LIKE 'p_t%';
```

/*Na coluna carga temos varios cursos com carga horaria de 40 horas.O distinct vai pegar todos os cursos que tem carga horaria de 40 horas, ou seja, que tem carga horarias iguais e só vai mostrar uma vez essa carga horaria de 40 horas, */

```
SELECT DISTINCT carga from tb_cursos;
```

/*O distinct vai selecionar a "nacionalidade" das pessoas na tabela gafanhotos e vai mostrar apenas uma vez Exemplo:Pode ter nessa tabela 10 pessoas com a nacionalidade brasileira, mas só vai mostrar uma vez essa nacionalidade Brasileira*/

```
SELECT DISTINCT nacionalidade FROM tb_gafanhotos;
```

```
SELECT DISTINCT nacionalidade FROM tb_gafanhotos  
ORDER BY nacionalidade;
```

```
SELECT DISTINCT carga FROM tb_cursos  
ORDER BY carga;
```

/*Vai mostrar na tela quantos cursos tem cadastrado na base de dados da tabela cursos*/

```
SELECT COUNT(nome) FROM tb_cursos;
```

```
SELECT COUNT(idcurso) FROM tb_cursos;
```

/*Vai mostrar na tela quantos cursos com carga horaria maior que 40 tem na tabela cursos*/

```
SELECT COUNT(*) FROM tb_cursos WHERE carga > 40;
```

/*Vai mostrar na tela qual e o maior número de aulas entre os cursos que estão na tabela cursos(totaulas)*/

```
SELECT MAX(totaulas) from tb_cursos;
```

/*Vai mostrar na tela qual e a maior carga horaria entre os cursos registrados na tabela cursos*/

```
SELECT MAX(carga) FROM tb_cursos;
```