

PRINCIPAIS COMANDOS SQL

CRIAR UMA BASE DE DADOS (BANCO DE DADOS):

CREATE DATABASE IF NOT EXISTS locadora

CRIAR UMA TABELA:

```
CREATE TABLE filmes(  
id int(10) unsigned not null auto_increment,  
titulo varchar(80) not null,  
ano int(4) unsigned not null,  
diretor varchar(80) not null,  
primary key (id))
```

VISUALIZAR A ESTRUTURA DE UMA TABELA:

DESC filmes

INSERIR DADOS EM UMA TABELA:

```
INSERT INTO filmes VALUES ( , "Titulo1", 1989, "Diretor1"),  
( , "Titulo2", 2001, "Diretor2")
```

SELECT - CONSULTA

CONSULTAR TODOS OS CAMPOS:

SELECT * FROM filmes

CONSULTAR UM OU N CAMPOS:

SELECT id, titulo FROM filmes

CONSULTAR VALOR MÁX. DE UM CAMPO:

SELECT MAX(id) AS identificação FROM filmes

ORDENAR UMA CONSULTA ASCENDENTE:

```
SELECT id, titulo FROM filmes  
ORDER BY id ASC
```

ORDENAR UMA CONSULTA DESCENDENTE:

```
SELECT id, titulo FROM filmes  
ORDER BY id DESC
```

CONSULTAR CAMPOS COM VALORES ESTIPULADOS:

```
SELECT * FROM filmes  
WHERE id=2
```

AGRUPAR CONSULTAS POR CAMPOS:

```
SELECT id, titulo FROM filmes  
GROUP BY id
```

CONSULTAS COM 2 TABELAS:

```
SELECT vendas.id, vendas.produto, clientes.id, clientes.nome  
FROM vendas, clientes  
WHERE vendas.id = clientes.id
```

CONSULTA RENOMEANDO A TABELA, USE AS:

```
SELECT ve.id, ve.produto, cli.id, cli.nome  
FROM vendas AS ve, clientes AS cli  
WHERE ve.id = cli.id AND ve.produto = "detergente"
```

SE EXISTE MAIS DE UM ANIMAL DE ESTIMAÇÃO COM O MESMO DONO MAS QUEREMOS VER SOMENTE UM NOME DE DONO DE CADA VEZ, USE DISTINCT:

SELECT DISTINCT dono FROM animaisestimacao

SELECIONAR CAMPOS CUJOS VALORES COMECEM COM DETERMINADA LETRA, USE LIKE:

SELECT * FROM tabela WHERE nome LIKE "a%"

ALTER TABLE - ALTERAÇÕES COM A ESTRUTURA DA TABELA

RENOMEAR UMA TABELA

sp_rename tabela tabela2

ADICIONAR UMA COLUNA(CAMPO) CHAMADA d DO TIPO DOUBLE:

ALTER TABLE tabela ADD d DOUBLE

DESAZER COMANDO ANTERIOR (CRIAÇÃO CAMPO):

ALTER TABLE tabela DROP d

Para alterar o tipo da coluna a de INTEGER para TINYINT NOT NULL

ALTER TABLE tabela ALTER COLUMN a TINYINT NOT NULL

ALTERAR A COLUNA d COMO CHAVE PRIMÁRIA:

ALTER TABLE tabela ADD PRIMARY KEY (d)

PARA REMOVER A COLUNA c:

ALTER TABLE tabela DROP COLUMN d

REPLACE

REPLACE funciona exatamente como o INSERT, exceto que se um registro antigo na tabela tem o mesmo valor que um novo registro em um índice UNIQUE ou PRIMARY KEY, o registro antigo é deletado antes que o novo registro seja inserido.

```
REPLACE INTO tabela (col1, col2, ...)  
VALUES (exp1, exp2, ...)
```

```
REPLACE INTO tabela (col1, col2, ...)  
SELECT ...
```

```
REPLACE INTO tabela (col1, col2, ...)  
SET nome_coluna=expressão,  
nome_coluna=expressão,...
```

UPDATE

Atribuições **UPDATE** atualizam o valor do campo com um novo valor e são avaliadas da esquerda para a direita. Por exemplo, a seguinte instrução dobra a coluna **age** e então a incrementa:

UPDATE dadospessoais SET idade=idade*2, idade=idade+1

UPDATE tabela SET nasc=26/02/82 WHERE id=2

```
UPDATE itens,meses SET itens.preço=meses.preço  
WHERE itens.id=meses.id
```

DELETE

DELETE FROM tabela WHERE codigo=2