

Resumão SQL

☰ Week	
📅 Date	
☰ Notes	Todos os comandos vistos em aula. <i>Por Giovana.</i>

CONSULTAR | LER | READ

```
SELECT colunas FROM nome_tabela;

SELECT coluna1, coluna2, coluna3, colunaN FROM tabela;

SELECT * FROM tabela;
```

COMENTÁRIO

```
/* Mostrar todos os dados de bebes */
SELECT * FROM bebes;

/* Mostrar nome e data de nascimento das mães */
SELECT nome, dt_nasc FROM maes;

/* Mostrar nome, crm e especialidade dos médicos */
SELECT nome, crm, especialidade FROM medicos;

/* Muda a ordem das colunas, cria nova tabela em tempo de execução, ela não altera os dados */
SELECT especialidade, nome, crm FROM medicos;
```

INSERÇÃO DE VALORES

```
INSERT INTO maes (id_mae, nome, endereco, telefone, dt_nasc)
VALUES (10, 'Rosário Silva', 'Rua zzz', '(11)900000000', '21/06/1994');

INSERT INTO maes VALUES (10, 'Rosário Silva', 'Rua zzz', '(11)900000000', '21/06/1994');

/* Caso não queira preencher um campo NULL OU '' */
INSERT INTO maes VALUES (NULL, 'Rosário Silva', '', '(11)900000000', '21/06/1994');

/* Preencher vários campos */
INSERT INTO maes VALUES
(NULL, 'Rosário Silva', '', '(11)900000000', '21/06/1994'),
(NULL, 'Joana Santos', '', '(11)911111111', '21/06/1994'),
(NULL, '', '', '(11)900000000', '21/06/1994'),
(NULL, '', '', '(11)900000000', '21/06/1994');
```

DELETAR DADOS (LINHAS)

```
/* dessa forma apaga todos os dados da tabela */
DELETE FROM nome_da_tabela;

DELETE FROM maes;

/* melhor forma: */
DELETE FROM maes WHERE id_mae = 4;

DELETE FROM maes WHERE nome = 'Tereza Cristina';

DELETE FROM maes WHERE id_mae = 4;

DELETE FROM maes WHERE id_mae > 5 AND id_mae < 10;

DELETE FROM maes WHERE id_mae = 4 OR id_mae = 5;
```

ATUALIZAR DADOS

```
/* dessa forma altera todos os dados da coluna */
UPDATE nome_da_tabela
SET coluna1 = novo_valor1, coluna2 = novo_valor2, colunaN = novo_valorN
```

```

UPDATE maes
SET endereco = 'Rua Rouxinol';

UPDATE maes
SET endereco = 'Rua Rouxinol', telefone = '(88)900000000';

/* melhor forma: */
UPDATE nome_da_tabela
SET coluna1 = novo_valor1, coluna2 = novo_valor2, colunaN = novo_valorN
WHERE condicao;

UPDATE maes
SET endereco = 'Rua Rouxinol'
WHERE id_mae = 4;

```

VARIAÇÕES DO SELECT

```

/* SELECT DISTINCT */
SELECT DISTINCT cidade
FROM compras;

```

```

/* ORDER BY */
SELECT * FROM maes ORDER BY nome;

SELECT * FROM maes ORDER BY nome DESC;

```

```

/* MIN E MAX */
SELECT MIN(alt_nasc) FROM bebes;

SELECT MAX(alt_nasc) FROM bebes;

```

```

/* COUNT */
SELECT COUNT(id_bebe) FROM bebes;

```

```

/* SUM */
SELECT SUM(alt_nasc) FROM bebes;

```

```

/* AVG */
SELECT AVG(alt_nasc) FROM bebes;

```

```

/* LIKE */
SELECT colunas
FROM nome_tabela
WHERE condicao LIKE 'algum_padrao';

SELECT *
FROM maes
WHERE nome LIKE 'Maria%';

SELECT *
FROM maes
WHERE nome LIKE '%Maria%';

SELECT *
FROM maes
WHERE nome LIKE '%Silva';

SELECT *
FROM maes
WHERE nome NOT LIKE '%Mari_%';

SELECT *
FROM maes
WHERE nome LIKE '%Silvi[ao]';

```

```
/* APELIDOS - ALIASES (AS) */
SELECT nome, dt_nasc, AVG(alt_nasc) AS media_altura, SUM(peso_nasc) AS soma_peso
FROM bebes;
```

```
/* BETWEEN */
SELECT nome, peso
FROM bebes
WHERE peso_bebe BETWEEN 3 AND 4;

SELECT nome, peso
FROM bebes
WHERE dt_nasc BETWEEN '2020-01-01' AND '2022-11-30';

SELECT nome, peso
FROM bebes
WHERE dt_nasc NOT BETWEEN '2020-01-01' AND '2022-11-30';
```

CRIAR BANCO DE DADOS

```
CREATE DATABASE nome_do_banco;
```

DEFINIR QUAL BANCO EU QUERO UTILIZAR

```
USE nome_do_banco;
```

VISUALIZAR QUAIS BANCOS DE DADOS EXISTEM

```
SHOW DATABASES;
```

FAZER BACKUP DO BANCO DE DADOS

```
BACKUP DATABASE nome_do_banco
TO DISK = 'url'
```

EXCLUIR BANCO DE DADOS

```
DROP DATABASE nome_do_banco;
```

CRIAR TABELAS

```
CREATE TABLE nome_da_tabela (
    atributo1 tipo,
    atributo2 tipo,
    .
    .
    atributoN tipoN
);
```

VISUALIZAR TABELAS QUE EXISTEM

```
SHOW TABLES;
```

VERIFICAR SCHEMA DA TABELA

```
DESCRIBE nome_da_tabela;
```

EXCLUIR TABELA

```
DROP TABLE nome_da_tabela;
```

EXCLUIR OS DADOS DA TABELA

```
TRUNCATE TABLE nome_da_tabela;
```

ALTERAR AS TABELAS

```
ALTER TABLE nome_da_tabela

/* ADICIONAR UMA NOVA COLUNA */
ALTER TABLE nome_da_tabela
ADD nome_da_coluna tipo;

/* EXCLUIR UMA COLUNA */
ALTER TABLE nome_da_tabela
DROP COLUMN nome_da_coluna;

/* MODIFICAR TIPO DE DADOS DE UMA COLUNA - SQL SERVER */
ALTER TABLE nome_da_tabela
ALTER COLUMN nome_da_coluna novo_tipo;

/* MODIFICAR TIPO DE DADOS DE UMA COLUNA - MYSQL */
ALTER TABLE nome_da_tabela
MODIFY COLUMN nome_da_coluna novo_tipo;

/* MODIFICAR TIPO DE DADOS DE UMA COLUNA - ORACLE */
ALTER TABLE nome_da_tabela
MODIFY nome_da_coluna novo_tipo;
```

CONSTRAINT

```
CREATE TABLE maes (
  id_mae INT AUTO_INCREMENT,
  nome VARCHAR(200) NOT NULL,
  endereco VARCHAR(200),
  telefone VARCHAR(20),
  dt_nasc DATE NOT NULL,
  CONSTRAINT pk_idMae PRIMARY KEY (id_mae)
);

/* OUTRA FORMA */

CREATE TABLE maes (
  id_mae INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(200) NOT NULL,
  endereco VARCHAR(200),
  telefone VARCHAR(20),
  dt_nasc DATE NOT NULL
);
```

Criando Databases e Tabelas

```
SHOW DATABASES;
```

```
CREATE DATABASE hospital;
```

```
USE hospital;
```

```
SHOW TABLES;
```

```
CREATE TABLE maes (
  id_mae INT AUTO_INCREMENT,
  nome VARCHAR(100) NOT NULL,
```

```
    endereco VARCHAR(200),
    telefone VARCHAR(50),
    dt_nasc DATE NOT NULL,
    CONSTRAINT pk_id_mae PRIMARY KEY(id_mae)
);
```

```
DESCRIBE maes;
```

```
CREATE TABLE medicos (
    id_medico INT AUTO_INCREMENT,
    crm VARCHAR(20) NOT NULL,
    nome VARCHAR(100) NOT NULL,
    especialidade VARCHAR(100) NOT NULL,
    CONSTRAINT pk_id_medico PRIMARY KEY(id_medico)
);
```

```
DESCRIBE medicos;
```

```
CREATE TABLE medicos_celular (
    id INT AUTO_INCREMENT,
    numero_celular VARCHAR(20) NOT NULL,
    id_medico INT,
    CONSTRAINT pk_id_medicos_celular PRIMARY KEY(id),
    CONSTRAINT fk_medicos_celular FOREIGN KEY(id_medico) REFERENCES medicos(id_medico)
);
```

```
DESCRIBE medicos_celular;
```

```
SHOW DATABASES;
```

```
CREATE DATABASE hospital;
```

```
USE hospital;
```

```
SHOW TABLES;
```

```
CREATE TABLE bebes (
    id_bebe INT AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    dt_nasc DATE NOT NULL,
    peso_nasc FLOAT NOT NULL,
    alt_nasc INT NOT NULL,
    id_medico INT,
    id_mae INT,
    CONSTRAINT pk_id_bebe PRIMARY KEY(id_bebe),
    CONSTRAINT fk_bebes_medicos FOREIGN KEY(id_medico) REFERENCES medicos(id_medico),
    CONSTRAINT fk_bebes_maes FOREIGN KEY(id_mae) REFERENCES maes(id_mae)
);
```

```
DESCRIBE maes;
```

```
INSERT INTO medicos VALUES (NULL, '65973624', 'Marcos Santana de Araújo', 'Obstetra');
```

```
SELECT * FROM medicos;
```

```
INSERT INTO medicos VALUES (NULL, '35998745', 'Eliana Martins de Lima', 'Obstetra');
```

```
SELECT id_medico, nome FROM medicos;
```

```
ALTER TABLE medicos  
DROP COLUMN especialidade;
```

```
ALTER TABLE bebes  
MODIFY COLUMN alt_nasc FLOAT;
```

```
ALTER TABLE bebes  
CHANGE alt_nasc altura_nasc INT;
```

```
ALTER TABLE medicos  
ADD email_medicos VARCHAR(100);
```

```
ALTER TABLE medicos  
ADD FOREIGN KEY(email_medicos) REFERENCES email_medicos(id_medico)
```

```
ALTER TABLE medicos  
ADD CONSTRAINT pk_id_medico PRIMARY KEY(id_medico);
```

```
INSERT INTO maes VALUES (NULL, 'Maria da Silva', 'Rua 12 de Agosto, 123', '(21)931110000', '1992-09-23'), (NULL, 'Juliana Santos', 'Ru
```

```
INSERT INTO medicos_celular VALUES (NULL, '(11)9.000000000', 1), (NULL, '(11)9.111111111', 2), (NULL, '(11)9.222222222', 2), (NULL, '(11)
```

```
INSERT INTO bebes VALUES (NULL, 'Joaquim da Silva', '2020-05-01', 3.250, 49, 2, 1), (NULL, 'Mariana Sales', '2021-07-26', 3.600, 51, 1
```

```
TRUNCATE TABLE bebes;
```

Consultas

```
SELECT * FROM medicos;
```

```
SELECT nome, dt_nasc FROM bebes;
```

```
SELECT nome, dt_nasc AS data_nascimento FROM bebes;
```

```
SELECT * FROM bebes WHERE alt_nasc > 50;
```

```
SELECT nome, telefone FROM maes;
```

```
SELECT * FROM medicos WHERE especialidade = 'Obstetra';
```

```

SELECT * FROM maes WHERE dt_nasc BETWEEN '1996-01-01' AND '1997-12-31';

/* outra forma */

SELECT * FROM maes WHERE YEAR(dt_nasc) = 1996 OR YEAR(dt_nasc) = 1997;

/* outra forma */
SELECT * FROM maes WHERE YEAR(dt_nasc) BETWEEN 1996 AND 1997;

```

```

SELECT * FROM bebes WHERE peso_nasc < 3.3;

```

JOIN

```

SELECT *
FROM tabela1 INNER JOIN tabela2
ON tabela1.atributo = tabela2.atributo

```

```

SELECT medicos.nome, medicos.crm, medicos_celular.numero_celular
FROM medicos INNER JOIN medicos_celular
ON medicos.id_medico = medicos_celular.id_medico
WHERE medicos.id_medico = 2;

```

```

/* forma enxuta */

```

```

SELECT m.nome, m.crm, mc.numero_celular
FROM medicos AS m INNER JOIN medicos_celular AS mc
ON m.id_medico = mc.id_medico
WHERE m.id_medico = 2;

```

```

SELECT b.nome AS nome_bebe, ma.nome AS nome_mae, m.nome AS nome_medico
FROM bebes AS b
INNER JOIN maes AS ma
ON b.id_mae = ma.id_mae
INNER JOIN medicos AS m
ON b.id_medico = m.id_medico;

```

Exercício 1:

```

CREATE DATABASE loja;

```

```

USE loja;

```

```

CREATE TABLE clientes (
id_cli INT AUTO_INCREMENT,
rg_cli VARCHAR(20) NOT NULL,
nome_cli VARCHAR(100) NOT NULL,
tel_cli VARCHAR(50),
rua VARCHAR(100),
num INT,
bairro VARCHAR(100),
cep VARCHAR(20),
CONSTRAINT pk_id_cli PRIMARY KEY(id_cli)
);

```

```

CREATE TABLE produtos (
id_prod INT AUTO_INCREMENT,
nome_prod VARCHAR(100) NOT NULL,
tipo_prod VARCHAR(100) NOT NULL,
preço_prod FLOAT NOT NULL,
estoque_prod INT,
CONSTRAINT pk_id_prod PRIMARY KEY(id_prod)
);

```

```
CREATE TABLE compras (
  id_compra INT AUTO_INCREMENT,
  data_compra DATE NOT NULL,
  valor_compra FLOAT NOT NULL,
  id_cli INT,
  CONSTRAINT pk_id_compra PRIMARY KEY(id_compra),
  CONSTRAINT fk_compra_cliente FOREIGN KEY(id_cli) REFERENCES clientes(id_cli)
);
```

```
CREATE TABLE compras_contem_produtos (
  id INT AUTO_INCREMENT,
  id_compra INT,
  id_prod INT,
  quantidade INT NOT NULL,
  CONSTRAINT pk_id_compra_produtos PRIMARY KEY(id),
  CONSTRAINT fk_ccp_compra FOREIGN KEY(id_compra) REFERENCES compras(id_compra),
  CONSTRAINT fk_ccp_produto FOREIGN KEY(id_prod) REFERENCES produtos(id_prod)
);
```

```
ALTER TABLE clientes
ADD cidade VARCHAR(100);
```

```
ALTER TABLE clientes
ADD estado VARCHAR(100);
```

```
ALTER TABLE compra
DROP CONSTRAINT fk_compras_clientes;
```

```
ALTER TABLE tabela
ADD CONSTRAINT fk_nome_da_constraint
FOREIGN KEY(atributo_chave)
REFERENCES tabela(atributo_chave)
ON UPDATE CASCADE
ON DELETE CASCADE;
```

```
INSERT INTO clientes
VALUES
(NULL,'42.515.012-4','Leonardo Diego das Neves','(68)9.8106-8326','Avenida Santos Dumont',160,'Centro','69934-970','Epitaciolândia','A
(NULL,'35.993.799-8','Valentina Jenniffer Livia','(64) 98941-7452','Avenida 11 de Novembro',729,'Setor Central','75619-970','Aloândia
(NULL,'14.594.456-6','Benedito Henry Teixeira','(88) 99690-4161','Rua Eduardo Mota',640,'Centro','62420-970','Chaval','CE'),
(NULL,'11.946.174-2','Mariah Jaqueline Aparício','(88) 3753-8770','Rua Santa Isabel',539,'São Miguel','63010-555','Juazeiro do Norte',
(NULL,'27.722.583-8','Alice Pietra Ramos','(31) 99404-8773','Rua Doutor Álvaro Lobo Leite',521,'Centro','36419-970','Lobo Leite','MG')
(NULL,'38.895.346-9','Mário Márcio Ruan Pinto','(13) 98485-6954','Rua Rubião Júnior',418,'Centro','11013-210','Santos','SP');
```

```
INSERT INTO compras
VALUES
(NULL,'2022-10-05',58.8,4),
(NULL,'2022-10-05',110,1),
(NULL,'2022-10-07',81.7,3),
(NULL,'2022-10-07',51,6);
```

```
INSERT INTO produtos
VALUES
(NULL,'Energético TNT 269ML','Bebidas',5.5,35),
(NULL,'Arroz Juliano Parboilizado','Arroz',4.5,150),
(NULL,'Arroz Juliano Branco','Arroz',4.2,122),
(NULL,'Coca Cola 1 litro','Bebidas',8.5,36),
(NULL,'Biscoito Recheado Bolão Morango','Doces',2.7,45),
(NULL,'Água Mineral sem Gás','Bebidas',2,266),
(NULL,'Chocolate Gutti Gutti 50g','Doces',1.9,61),
(NULL,'Suco de Laranja Laranjinha','Bebidas',3,16);
```

```
INSERT INTO compras_contem_produtos
VALUES
```



```
(NULL,1,3,10),
(NULL,1,5,4),
(NULL,1,8,2),
(NULL,2,2,20),
(NULL,2,6,10),
(NULL,3,1,4),
(NULL,3,3,10),
(NULL,3,7,3),
(NULL,3,8,4),
(NULL,4,4,6);
```

```
SELECT * FROM clientes;
```

```
SELECT rg_cli, nome_cli, tel_cli FROM clientes;
```

```
SELECT nome_prod, preço_prod FROM produtos;
```

```
SELECT nome_prod, preço_prod, estoque_prod
FROM produtos
WHERE tipo_prod = 'bebidas';
```

```
SELECT * FROM produtos WHERE preço_prod > 5;
```

```
SELECT SUM(estoque_prod) FROM produtos;
```

```
SELECT SUM(valor_compra) FROM compras;
```

```
SELECT * FROM produtos ORDER BY nome_prod;
```

```
SELECT cl.nome_cli, co.valor_compra
FROM clientes AS cl INNER JOIN compras AS co
ON cl.id_cli = co.id_cli
WHERE cl.nome_cli = 'Mariah Jaqueline Aparicio';
```

```
SELECT * FROM clientes
WHERE estado = 'SP' OR estado = 'CE';
```

```
SELECT * FROM clientes
WHERE estado = 'CE' AND bairro = 'centro';
```

```
SELECT * FROM produtos ORDER BY preço_prod;
```

```
SELECT p.nome_prod, cl.nome_cli, cl.rua, cl.num, cl.bairro, cl.cep, cl.cidade, cl.estado
FROM produtos AS p
INNER JOIN compras_contem_produtos AS ccp
ON p.id_prod = ccp.id_prod
INNER JOIN compras AS co
ON ccp.id_compra = co.id_compra
INNER JOIN clientes AS cl
ON cl.id_cli = co.id_cli
WHERE cl.nome_cli = 'Benedito Henry Teixeira';
```

```
SELECT COUNT(*) FROM clientes WHERE estado = 'CE';
```

```
SELECT SUM(estoque_prod) FROM produtos WHERE tipo_prod = 'bebidas';
```