

Laboratório de SQL - Parte I: Comandos básicos.

-- MySQL -----

```
CREATE TABLE Pessoas
(
  cod_pessoa INT AUTO_INCREMENT,
  nome       VARCHAR(40),
  telefone   VARCHAR(20),
  identidade  VARCHAR(20),
  sexo       CHAR(1),
  rua        VARCHAR(40),
  numero     VARCHAR(10),
  bairro     VARCHAR(40),
  cep        VARCHAR(8),
  cidade     VARCHAR(40),
  uf         CHAR(2),
  PRIMARY KEY(cod_pessoa)
);
```

```
SHOW tables;
```

```
DESCRIBE pessoas;
```

-- MySQL -----

-- POSTGRESQL -----

```
CREATE TABLE Pessoas
(
  cod_pessoa SERIAL PRIMARY KEY,
  nome       VARCHAR(40),
  telefone   VARCHAR(20),
  identidade  VARCHAR(20),
  sexo       CHAR(1),
  rua        VARCHAR(40),
  numero     VARCHAR(10),
  bairro     VARCHAR(40),
  cep        VARCHAR(8),
  cidade     VARCHAR(40),
  uf         CHAR(2)
);
```

```
SELECT table_name FROM information_schema.tables WHERE table_schema =
'public';
```

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE table_name = 'pessoas';
```

-- POSTGRESQL -----

```

-- sqlite3 ---
CREATE TABLE Pessoas
(
  cod_pessoa    INTEGER PRIMARY KEY AUTOINCREMENT,
  nome          VARCHAR(40),
  telefone      VARCHAR(20),
  identidade     VARCHAR(20),
  sexo         CHAR(1),
  rua           VARCHAR(40),
  numero        VARCHAR(10),
  bairro        VARCHAR(40),
  cep           VARCHAR(8),
  cidade        VARCHAR(40),
  uf            CHAR(2)
);

-- sqlite3 ----

--
-- INSERINDO DADOS NA TABELA PESSOAS
--

INSERT INTO Pessoas (nome, rua, numero, bairro,
                    cidade, uf, cep, telefone, identidade,
                    sexo)
VALUES ('Jose Francisco de Queiroz', 'Aquarius', '258',
        'Vila So Jos', 'SJC', 'MG',
        '12227320', '3911-7707', 'M-X.XXX.XXX', 'M');

INSERT INTO Pessoas (nome, rua, numero, bairro,
                    cidade, uf, cep, telefone, identidade,
                    sexo)
VALUES ('Eduardo Silva de Queiroz', 'Albino Sartori',
        '95', 'Vila Sao Jose', 'Ouro Preto', 'MG',
        '35400000', '3551-4243', 'M-X.23X.0X4', 'M');

INSERT INTO Pessoas (nome, rua, numero, bairro,
                    cidade, uf, cep, telefone, identidade,
                    sexo)
VALUES ('Maria Antonia de Oliveira', '36', '1020',
        'Centro', 'Ituiutaba', 'SP',
        '36400000', '3921-1248', 'M-19.239.013', 'F');

INSERT INTO Pessoas (nome, rua, numero, bairro,
                    cidade, uf, cep, telefone, identidade,
                    sexo)
VALUES ('Marcia Teffolo de Oliveira', 'Pandi Calgeras',

```

```
'580', 'Cabeas', 'Ouro Preto', 'MT', '35400000',  
'3551-1807', 'M-9.239.019', 'F');
```

```
INSERT INTO Pessoas (nome, rua, numero, bairro,  
                    cidade, uf, cep, telefone, identidade,  
                    sexo)  
VALUES ('Joaquim Queiroz da Silva', 'Goias', '58',  
        'Centro', 'Santa Vitoria', 'GO',  
        '12327320', '4911-7707', 'M-9.239.000', 'M');
```

```
INSERT INTO Pessoas  
VALUES (20, 'Maria Tereza Melo', '3561-7607', '11220333-6',  
        'M', 'Francisco Jose Longo', '180',  
        'Centro', '12227320', 'SJC', 'SP');
```

```
SELECT nome, cidade FROM pessoas;
```

```
SELECT * FROM pessoas WHERE sexo = 'M';
```

```
SELECT * FROM pessoas WHERE nome LIKE '%Queiroz%';
```

```
SELECT DISTINCT cidade FROM pessoas;
```

```
> Selecionar mulheres de Ouro Preto  
> Selecionar moradores da Regiao Sudeste
```

```
SELECT * FROM pessoas ORDER BY cidade;
```

```
SELECT * FROM pessoas ORDER BY cidade DESC;
```

```
SELECT * FROM pessoas  
WHERE cidade IN ('Ouro Preto', 'Ituiutaba')
```

```
UPDATE pessoas SET cidade = 'Sao Jose dos Campos' WHERE cidade = 'SJC';
```

```
DELETE FROM pessoas WHERE sexo = 'F';
```

```
>> delete from pessoas
```

```
-----  
-- MySQL -----
```

```
CREATE TABLE pedidos (  
    cod_pedido INT AUTO_INCREMENT,  
    nro_pedido INT NOT NULL,  
    cod_pessoa INT NOT NULL,  
    PRIMARY KEY(cod_pedido));
```

-- MySQL -----

-- POSTGRESQL ---

```
CREATE TABLE pedidos (  
  cod_pedido SERIAL PRIMARY KEY,  
  nro_pedido INT NOT NULL,  
  cod_pessoa INT NOT NULL);
```

-- POSTGRESQL ---

```
INSERT INTO pedidos VALUES  
(1,77895,3), (2,44678,3),(3,22456,1),(4,24562,1),(5,34764,15);
```

```
// selecione o nome das pessoas, sua cidade e seus pedidos  
SELECT pessoas.nome, pessoas.cidade, pedidos.nro_pedido  
FROM pessoas  
INNER JOIN pedidos  
ON pessoas.cod_pessoa=pedidos.cod_pessoa  
ORDER BY pessoas.nome;
```

```
SELECT pessoas.nome, pessoas.cidade, pedidos.nro_pedido  
FROM pessoas  
LEFT JOIN pedidos  
ON pessoas.cod_pessoa=pedidos.cod_pessoa  
ORDER BY pessoas.nome;
```

```
SELECT pessoas.nome, pessoas.cidade, pedidos.nro_pedido  
FROM pessoas  
RIGHT JOIN pedidos  
ON pessoas.cod_pessoa=pedidos.cod_pessoa  
ORDER BY pessoas.nome;
```

--- MySQL ---

```
CREATE TABLE compras (  
  item INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT NULL,  
  fornecedor CHAR(20) DEFAULT '' NOT NULL,  
  preco DOUBLE(16,2) DEFAULT '0.00' NOT NULL,  
  PRIMARY KEY(item, fornecedor));
```

--- MySQL ---

--- POSTGRESQL ---

```
CREATE TABLE compras (  
  item Integer DEFAULT '0000' NOT NULL,  
  fornecedor CHAR(20) DEFAULT '' NOT NULL,  
  preco real DEFAULT '0.00' NOT NULL,  
  PRIMARY KEY(item, fornecedor));
```

--- POSTGRESQL ---

```
INSERT INTO compras VALUES
(1,'Extra',3.45),(1,'Pao de Acucar',3.99),(2,'Extra',10.99),(3,'Pao de
Acucar',1.45),(3,'Wall-Mart',1.69),
(3,'Makro',1.25),(4,'Makro',19.95);
```

- Quantas compras foram feitas?

```
SELECT COUNT(*) FROM compras;
```

- Quantos itens diferentes foram comprados?

```
SELECT COUNT(DISTINCT item) FROM compras;
```

- Qual o item com maior código?

```
SELECT MAX(item) AS item FROM compras;
```

- Quanto foi gasto em compras?

```
SELECT SUM(preco) AS total FROM compras;
```

- Quanto foi pago para cada fornecedor?

```
SELECT fornecedor, SUM(preco) FROM compras GROUP BY fornecedor;
```

- Quais fornecedores receberam mais que 10?

```
SELECT fornecedor, SUM(preco) FROM compras
GROUP BY fornecedor
HAVING SUM(preco)>10;
```

- Encontre o número, fornecedor e preço do item mais caro.

```
SELECT item, fornecedor, preco
FROM compras
WHERE preco=(SELECT MAX(preco) FROM compras);
```

- Qual é o maior preço por item?

```
SELECT item, MAX(preco) AS preco
FROM vendas
GROUP BY item;
```

- Para cada item, encontre o(s) fornecedor(s) com o maior preço.

```
SELECT item, fornecedor, preco
FROM compras v1
WHERE preco=(SELECT MAX(v2.preco)
              FROM compras v2
              WHERE v1.item = v2.item);
```

```
UPDATE compras
```

```
    SET preco = preco+ preco*0.1
```

```
    WHERE preco > 5;
```

```
create table focos
```

```

(
nro    integer PRIMARY KEY,
lat      real,
lon real,
latGMS   VARCHAR(20),
lonGMS   VARCHAR(20),
data     VARCHAR(20),
hora     VARCHAR(20),
satelite   VARCHAR(20),
municipio VARCHAR(40),
estado   VARCHAR(2),
vegetacao VARCHAR(50),
susctibilidade VARCHAR(50),
precipitacao    real,
numdiasemchuva  integer,
risco           real,
persistencia real
);

alter table focos
alter column municipio TYPE varchar(50);

copy focos from 'c:/fire_spot_sp.csv' CSV;

```

```
//----
```

```

CREATE TABLE cidades (
nome    varchar(80) primary key,
x       real,
y       real
);

```

```
--- MySQL
```

```

CREATE TABLE tempo (
cidade    varchar(80) references cidades(nome),
temp_bx   int,
temp_al   int,
prcp      real,
data      date
);

```

```
-- Mysql
```

```
--- SQLite
```

```

CREATE TABLE tempo (
cidade    varchar(80),
temp_bx   int,
temp_al   int,

```

```

        prcp            real,
        data            date,
        FOREIGN KEY(cidade) REFERENCES cidades(nome));

--- SQLite

INSERT INTO cidades VALUES ('Sao Jose dos Campos',-23.2, -43.6);
INSERT INTO cidades VALUES ('Campos do Jordao',-22.12, -42.5);

INSERT INTO tempo VALUES ('Sao Jose dos Campos', 13, 21, 0.25,
'2010-06-13');
INSERT INTO tempo VALUES ('Sao Jose dos Campos', 11, 21, 0.30,
'2010-06-14');
INSERT INTO tempo VALUES ('Sao Jose dos Campos', 11, 21, 0.27,
'2010-06-15');
INSERT INTO tempo VALUES ('Sao Jose dos Campos', 13, 21, 0.25,
'2010-06-16');
INSERT INTO tempo VALUES ('Sao Jose dos Campos', 19, 24, 0.22,
'2010-06-17');

INSERT INTO tempo VALUES ('Campos do Jordao', 4, 11, 0.21, '2010-06-13');
INSERT INTO tempo VALUES ('Campos do Jordao', 0, 9, 0.32, '2010-06-14');
INSERT INTO tempo VALUES ('Campos do Jordao', -1, 8, 0.17, '2010-06-15');
INSERT INTO tempo VALUES ('Campos do Jordao', 5, 11, 0.15, '2010-06-16');
INSERT INTO tempo VALUES ('Campos do Jordao', 3, 14, 0.22, '2010-06-17');

INSERT INTO tempo VALUES ('Taubate', 9, 22, 0.11, '2010-06-13');

// consertar!
INSERT INTO cidades VALUES ('Taubate',-21.12, -42.3);
INSERT INTO tempo VALUES ('Taubate', 11, 25, 0.12, '2010-06-14');
INSERT INTO tempo VALUES ('Taubate', 12, 27, 0.17, '2010-06-15');
INSERT INTO tempo VALUES ('Taubate', 7, 24, 0.25, '2010-06-16');
INSERT INTO tempo VALUES ('Taubate', 8, 24, 0.12, '2010-06-17');

SELECT cidade, (temp_bx+temp_al)/2 AS temp_med, data FROM tempo;

SELECT *
FROM tempo, cidades
WHERE cidade = nome;

SELECT tempo.cidade, tempo.temp_bx, tempo.temp_al,
        tempo.prcp, tempo.data, cidades.x, cidades.y
        FROM tempo, cidades
        WHERE cidades.nome = tempo.cidade;

SELECT *
        FROM tempo INNER JOIN cidades ON (tempo.cidade = cidades.nome);

```

```
SELECT *  
    FROM tempo LEFT OUTER JOIN cidades ON (tempo.cidade = cidades.nome);
```

```
SELECT t1.cidade, t1.temp_bx AS baixa, t1.temp_al AS alta,  
    t2.cidade, t2.temp_bx AS baixa, t2.temp_al AS alta  
    FROM tempo t1, tempo t2  
    WHERE t1.temp_bx < t2.temp_bx  
    AND t1.temp_al > t2.temp_al;
```

```
SELECT max(temp_bx) FROM tempo;
```

- Abaixe 2 graus em todas as temperaturas coletadas depois de 2010-06-16;
- Selecione as leituras mais baixas de precipitação para cada cidade
- Selecione a precipitação mais baixa por cidade, a partir de um mínimo de 0.25;

```
CREATE VIEW myview AS  
    SELECT cidade, temp_bx, temp_al, prcp, data, x, y  
    FROM tempo, cidades  
    WHERE cidade = nome;
```

```
SELECT * FROM myview;
```