



Sumarizando informações

As cláusulas que vimos anteriormente são os tijolos elementares para realizarmos quaisquer consultas em tabelas num banco de dados relacional. Embora individualmente elas sejam bastante simples, sua combinação é justamente o que torna o SQL tão expressivo e poderoso.

Exemplo:

Vamos criar o database "bd_papelaria" e as tabelas "tb_produto" e "tb_entrada_produto". A tabela de "tb_produto" vai conter o cadastro dos produtos que a papelaria vende.

```
/*Estrutura do database bd_papelaria*/
```

```
CREATE DATABASE bd_papelaria
WITH
  OWNER = letscode
  ENCODING = 'UTF8'
  CONNECTION LIMIT = -1;
```

```
/*Estrutura da tabela tb_produto*/
```

```
CREATE TABLE tb_produto (
  id SERIAL NOT NULL,
  status CHAR(1) NOT NULL DEFAULT 'A',
  descricao VARCHAR(50) NULL DEFAULT NULL,
  estoque_minimo INT NULL DEFAULT NULL,
  estoque_maximo INT NULL DEFAULT NULL,
  PRIMARY KEY (id));
```

```
/*Inserindo 20 registros na tabela tb_produto
```

```
Campo "status" indica se o cadastro está ativo "A" ou inativo "I"*/
```

```
INSERT INTO tb_produto VALUES (0001, 'A', 'Lapis', 10, 20);
INSERT INTO tb_produto VALUES (0002, 'A', 'Lapis de cor', 12, 60);
INSERT INTO tb_produto VALUES (0003, 'A', 'Lapis de cera', 12, 50);
INSERT INTO tb_produto VALUES (0004, 'A', 'Marcadores', 10, 50);
```

```

INSERT INTO tb_produto VALUES (0005, 'A', 'Esferograficas (azul, vermelha,
preta e verde)', 7, 30);
INSERT INTO tb_produto VALUES (0006, 'A', 'Etiqueta', 8, 100);
INSERT INTO tb_produto VALUES (0007, 'A', 'Borracha', 20, 70);
INSERT INTO tb_produto VALUES (0008, 'A', 'Caderno pautado', 30, 50);
INSERT INTO tb_produto VALUES (0009, 'A', 'Caderno quadriculado', 30, 50);
INSERT INTO tb_produto VALUES (0010, 'A', 'Caderno de musica', 16, 50);
INSERT INTO tb_produto VALUES (0011, 'A', 'Calculadora', 20, 30);
INSERT INTO tb_produto VALUES (0012, 'A', 'Regua de 20 cm', 13, 40);
INSERT INTO tb_produto VALUES (0013, 'I', 'Compasso 8', 40);
INSERT INTO tb_produto VALUES (0014, 'I', 'Giz de cera', 9, 30);
INSERT INTO tb_produto VALUES (0015, 'I', 'Tesoura', 11, 20);
INSERT INTO tb_produto VALUES (0016, 'I', 'Estojo', 21, 100);
INSERT INTO tb_produto VALUES (0017, 'I', 'Papel Manteiga', 15, 50);
INSERT INTO tb_produto VALUES (0018, 'I', 'Mochila', 6, 100);
INSERT INTO tb_produto VALUES (0019, 'I', 'T-shirt', 25, 120);
INSERT INTO tb_produto VALUES (0020, 'I', 'Corretivo', 16, 30);

```

Na tabela **tb_entrada_produto** serão gravadas todas as compras de produtos efetuadas para a papelaria.

```
/*Estrutura da tabela tb_entrada_produto*/
```

```

CREATE TABLE tb_entrada_produto (
id SERIAL NOT NULL,
id_produto INT NULL DEFAULT NULL,
qtde INT NULL DEFAULT NULL,
valor_unitario DECIMAL(9,2) NULL DEFAULT '0.00',
data_entrada DATE NULL DEFAULT NULL,
PRIMARY KEY (id));

```

```
/*Inserindo 20 registros na tabela tb_entrada_produto*/
```

```

INSERT INTO tb_entrada_produto VALUES (0001, 0001, 20, '1.00', '2000-12-31');
INSERT INTO tb_entrada_produto VALUES (0002, 0002, 60, '1.00', '2021-01-01');
INSERT INTO tb_entrada_produto VALUES (0003, 0003, 50, '2.00', '2021-01-31');
INSERT INTO tb_entrada_produto VALUES (0004, 0004, 50, '5.00', '2021-01-31');
INSERT INTO tb_entrada_produto VALUES (0005, 0005, 30, '10.00', '2021-01-
22');
INSERT INTO tb_entrada_produto VALUES (0006, 0006, 100, '8.00', '2021-01-
13');
INSERT INTO tb_entrada_produto VALUES (0007, 0007, 70, '9.00', '2021-03-31');
INSERT INTO tb_entrada_produto VALUES (0008, 0008, 50, '17.00', '2021-03-
12');
INSERT INTO tb_entrada_produto VALUES (0009, 0009, 50, '9.00', '2021-03-09');
INSERT INTO tb_entrada_produto VALUES (0010, 0010, 50, '3.00', '2021-03-31');

```

```

INSERT INTO tb_entrada_produto VALUES (0011, 0011, 30, '2.00', '2021-02-31');
INSERT INTO tb_entrada_produto VALUES (0012, 0012, 40, '5.00', '2021-02-10');
INSERT INTO tb_entrada_produto VALUES (0013, 0013, 40, '4.00', '2021-02-31');
INSERT INTO tb_entrada_produto VALUES (0014, 0014, 30, '9.00', '2021-02-10');
INSERT INTO tb_entrada_produto VALUES (0015, 0015, 20, '3.00', '2021-02-13');
INSERT INTO tb_entrada_produto VALUES (0016, 0016, 100, '7.00', '2021-01-31');
INSERT INTO tb_entrada_produto VALUES (0017, 0017, 50, '2.00', '2021-01-25');
INSERT INTO tb_entrada_produto VALUES (0018, 0018, 100, '3.00', '2021-03-31');
INSERT INTO tb_entrada_produto VALUES (0019, 0019, 120, '8.00', '2021-03-30');
INSERT INTO tb_entrada_produto VALUES (0020, 0020, 30, '6.00', '2021-02-31');

```

A tabela **tb_estoque** é somente o resultado das ações de compra e venda de produtos.

```
/*Estrutura da tabela tb_estoque*/
```

```

CREATE TABLE tb_estoque (
id SERIAL NOT NULL,
id_produto INT NULL DEFAULT NULL,
qtde INT NULL DEFAULT NULL,
valor_unitario DECIMAL(9,2) NULL DEFAULT '0.00',
PRIMARY KEY (id));

```

```
/*Inserindo 20 registros na tabela tb_estoque*/
```

```

INSERT INTO tb_estoque VALUES (0001, 0001, 20, '1.00');
INSERT INTO tb_estoque VALUES (0002, 0002, 60, '1.50');
INSERT INTO tb_estoque VALUES (0003, 0003, 50, '2.00');
INSERT INTO tb_estoque VALUES (0004, 0004, 50, '5.00');
INSERT INTO tb_estoque VALUES (0005, 0005, 30, '10.00');
INSERT INTO tb_estoque VALUES (0006, 0006, 100, '8.00');
INSERT INTO tb_estoque VALUES (0007, 0007, 70, '9.00');
INSERT INTO tb_estoque VALUES (0008, 0008, 50, '17.00');
INSERT INTO tb_estoque VALUES (0009, 0009, 50, '9.00');
INSERT INTO tb_estoque VALUES (0010, 0010, 50, '3.00');
INSERT INTO tb_estoque VALUES (0011, 0011, 30, '2.00');
INSERT INTO tb_estoque VALUES (0012, 0012, 40, '5.00');
INSERT INTO tb_estoque VALUES (0013, 0013, 40, '4.00');
INSERT INTO tb_estoque VALUES (0014, 0014, 30, '9.00');
INSERT INTO tb_estoque VALUES (0015, 0015, 20, '3.00');
INSERT INTO tb_estoque VALUES (0016, 0016, 100, '7.00');
INSERT INTO tb_estoque VALUES (0017, 0017, 50, '2.00');
INSERT INTO tb_estoque VALUES (0018, 0018, 100, '3.00');

```

```
INSERT INTO tb_estoque VALUES (0019, 0019, 120, '8.00');
INSERT INTO tb_estoque VALUES (0020, 0020, 30, '6.00');
```

Na tabela **tb_saida_produto** serão gravadas todas as saídas (Vendas) de produtos.

```
/*Estrutura da tabela tb_saida_produto*/
```

```
CREATE TABLE tb_saida_produto (
  id SERIAL NOT NULL,
  id_produto INT NULL DEFAULT NULL,
  qtde INT NULL DEFAULT NULL,
  data_saida DATE NULL DEFAULT NULL,
  valor_unitario DECIMAL(9,2) NULL DEFAULT '0.00',
  PRIMARY KEY (id));
```

```
/*Inserindo 10 registros na tabela tb_saida_produto*/
```

```
INSERT INTO tb_saida_produto VALUES (0001, 0011, 10, '2021-05-01', '2.00');
INSERT INTO tb_saida_produto VALUES (0002, 0012, 20, '2021-05-01', '5.00');
INSERT INTO tb_saida_produto VALUES (0003, 0013, 40, '2021-05-01', '4.00');
INSERT INTO tb_saida_produto VALUES (0004, 0014, 04, '2021-05-01', '9.00');
INSERT INTO tb_saida_produto VALUES (0005, 0015, 05, '2021-05-01', '3.00');
INSERT INTO tb_saida_produto VALUES (0006, 0016, 100, '2021-05-01', '7.00');
INSERT INTO tb_saida_produto VALUES (0007, 0017, 30, '2021-05-01', '2.00');
INSERT INTO tb_saida_produto VALUES (0008, 0018, 20, '2021-05-01', '3.00');
INSERT INTO tb_saida_produto VALUES (0009, 0019, 05, '2021-05-01', '8.00');
INSERT INTO tb_saida_produto VALUES (0010, 0020, 04, '2021-05-01', '6.00');
```

Relatórios e gráficos são agregados de informações, precisamos aprender a juntar os dados em tabelas que sumariem esses dados. Contando, somando, tomando a média, entre outros.

Vamos aprender tais ações utilizando as tabelas **tb_produto**, **tb_entrada_produto**, **tb_estoque** e **tb_saida_produto**.

- **MIN() MAX()**

Começamos mostrando como obter os menores e os maiores valores de determinada coluna, de modo geral, ou em relação a outra coluna.

Para obter o produto com valor mínimo de uma coluna, usamos:

```
SELECT MIN(valor_unitario)
FROM tb_entrada_produto;
```

Para obter o produto com valor máximo de uma coluna, usamos:

```
SELECT MAX(valor_unitario)
FROM tb_entrada_produto;
```

- **COUNT()**

Podemos contar todas as linhas de uma tabela de um modo bem simples:

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

- **SUM()**

Para somar todos os valores por quantidade, usamos o SUM(). Abaixo, somamos os preços de todos os produtos da tabela por categoria.

```
SELECT qtde,
       SUM(valor_unitario)
FROM tb_saida_produto
GROUP BY 1;
```

- **AVG()**

Com frequência também precisamos tomar a média para cada conjunto de quantidade de produto na nossa base. Para tanto, temos o AVG(). A seguir, calculamos a média de quantidade de produtos para cada valor unitário:

```
SELECT valor_unitario,
       AVG(qtde)
FROM tb_entrada_produto
GROUP BY 1;
```

- **HAVING**

E se quisermos filtrar os resultados de uma query pelas contagens ou somas calculados nessa própria query? O WHERE não vai funcionar, porque ele não é capaz de filtrar resultados de funções agregadoras, como COUNT, MIN ou SUM. Para isto, existe o HAVING.

Abaixo ilustramos seu uso como filtro: após contarmos quantos itens há em cada valor unitário, exibiremos apenas aquelas linhas de valor unitário que tenham, associadas a si, mais do que 2 itens ao menos:

```
SELECT valor_unitario,
       COUNT(*)
FROM tb_estoque
GROUP BY 1
HAVING COUNT(*) > 2;
```

