

# Criando, Alterando e Apagando registros com SELECT.

Criando tabela

```
create table produtos2 (  
select * from produtos  
where Quantidade >= 10)
```

Inserir registros em uma tabela

```
insert into produtos2 (  
select id_prod, Nome, Valor, Quantidade, id_cad from produtos);
```

Atualizar dados de uma tabela a partir de select da outra.

```
update produtos2 set Nome = 'Abacaxi'  
where Nome in (select Nome from produtos);
```

Apagando Registros de uma tabela a partir de consulta de outra tabela.

```
delete from produtos2  
where Nome in (select Nome from produtos);
```

# Funções de Agregação,String,Aritméticas, Calculos,Data

No MySQL, existem várias funções nativas que nos possibilitam fazer diversas operações, dentre elas: realizar cálculos, manipular strings, trabalhar com datas, realizar opções lógicas, extrair informações dos nossos registros etc. Elas estão divididas nos seguintes tipos: numéricas, lógica, manipulação de string e funções de data e hora.

## Função de Agregação

As funções de agregação são responsáveis por agrupar vários valores e retornar somente um único para um determinado grupo.

```
select Quantidade, Nome_Cliente Prod_Vendidos  
from produtos, cadastro  
where cadastro.id_cad = produtos.id_cad  
order by Quantidade;
```

Utilizando Group By para evitar repetição dos dados.

```
insert into produtos (Nome, Valor, Quantidade, id_cad)  
Values('Laranja', 3.50, 15, 1);  
  
SELECT * FROM produtos;
```

```
select Nome, Nome_Cliente Prod_Vendidos  
from cadastro, produtos  
where cadastro.id_cad = produtos.id_cad  
group by Nome_Produtos  
order by Nome_Produtos;
```

## **Count**

O count pode ser usado apenas para contar a quantidade de registro em uma tabela.

```
select count(*) from produtos;
```

## **Having Count**

```
SELECT Nome_Cliente, Nome, Count(Nome)
```

```
FROM produtos, cadastro
```

```
where cadastro.id_cad = produtos.id_cad
```

```
GROUP BY Nome_Produtos
```

```
HAVING COUNT(Nome) < 3
```

## **MAX e MIN**

Mostrar Valores Maximos ou Minimos de um registro.

```
select max(Quantidade) QT_PROD from produtos
```

```
select min(Quantidade) QT_PROD from produtos
```

```
select min(Quantidade) QT_PROD_MIN, max(Quantidade) QT_PROD_MAX  
from produtos
```

## **SUM**

Calcula o valor total dos registros em um campo.

```
Select sum(Quantidade) Quantidade_Total, sum(Valor) Valor_TOTAL from  
produtos
```

## **AVG**

Cálculo de Média dos Valores de Campos

```
select format(avg(Valor),2) from produtos;
```

```
select avg(Valor) from produtos;
```

## **Função de String**

As funções de string (caracteres) podem ser utilizadas para modificar os dados, no que diz respeito aos valores selecionados. Podemos manipular os valores de texto.

### **Substr e length**

A função SUBSTRING na linguagem SQL é utilizada para obter uma parte dos dados armazenados.

A função LENGTH em SQL é utilizada para obter o comprimento de uma cadeia.

```
select Quantidade, Nome from produtos
```

```
where substr(Quantidade,1,40) = '30'
```

```
and length(Nome) > 3;
```

```
select Nome from produtos
```

```
where substr(Nome,1,6) = 'melanc'
```

## **Concat e concat\_ws**

Concat combina e junta resultados de campos distintos.

CONCAT\_WS junta valores com um separador qualquer. Isso evita colocarmos um espaço a cada campo concatenado.

```
select concat(Nome_Cliente,' Gosta De: ', Nome) Gosto_Frutas from  
cadastro, produtos  
where cadastro.id_cad = produtos.id_cad  
order by Nome_Cliente;
```

```
select concat_ws(';','Nome,Quantidade,Valor) from produtos  
where Nome like 'L%';
```

## **Lcase e Ucase**

Lcase retorna os registros em letras minúsculas.

Ucase retorna os registros em letras maiúsculas.

```
select lcase(Nome_Cliente) from cadastro;  
select ucase(Nome_Cliente) from cadastro;
```

## FUNÇÕES DE CÁLCULOS E OPERADORES ARITMÉTICOS

Funções de cálculos, como a descrição já diz, são utilizadas para realizar operações de cálculos. É de grande utilidade ter essas funções, pois, assim, não é preciso realizá-los usando apenas operadores comuns. Por exemplo, temos desde uma função para realizar o cálculo da raiz quadrada até uma que retorna a tangente de Pi!.

### Round

Retorna um número inteiro arredondado para um valor numérico. Utilizado em campos do tipo float.

```
select round('3213.34242',2) from dual;
```

### Truncate

É a opção de utilizar uma função que vai truncar as casas decimais, ou seja, omiti-las.

```
select truncate(max(Valor),0) maior_venda from produtos;
```

### SQRT

Retorna o resultado da raiz quadrada de um valor informado

```
select sqrt(4) Raiz_Quadrada;
```

## **PI**

Retorna valor de PI

```
select pi();
```

## **Cálculos Matemáticos: Adição, Subtração, Multiplicação, Divisão.**

### **Multiplicação**

```
select (Quantidade * Valor) multiplicação from produtos  
where id_prod = 4;
```

### **Adição**

```
select (Quantidade + Valor) Adição from produtos  
where id_prod = 4;
```

### **Subtração**

```
select (Quantidade - Valor) Subtração from produtos  
where id_prod = 4;
```

### **Divisão**

```
select (Quantidade / Valor) Divisão from produtos  
where id_prod = 4;
```

## **FUNÇÕES DE DATA**

Ela trabalha com dados de datas, formato de datas de um banco de dados e informações de dados temporais.

## **Curdate**

Trás a data atual

```
select curdate();
```

Retornar data e hora atual

```
select now();
```

```
select sysdate();
```

Retornar Hora Atual

```
select curtime();
```

Intervalo de Datas

```
select datediff('2018-01-15 23:59:59','2017-04-23');
```

Adicionar dias a uma data

```
select date_add('2019-02-01', interval 60 day);
```

Retornar dia da Semana

```
select dayname('2016-10-21');
```

Retornar dia do Mês

```
select dayofmonth('2019-06-27');
```

Extrair ano de uma data

```
select extract(year from '2020-09-21');
```

Retornar ultimo dia do mês

```
select last_day('2017-08-24');
```

Formatar padrão de data

```
select date_format('2015-11-04',get_format(date,'EUR'));
```



```
select str_to_date('05.05.2016',get_format(date,'USA'));
```

## **Uso de IF e CASE em SQL**

Testar a Quantidade de Frutas em Estoque e trazer um resultado baixo ou alto

```
select Nome,  
  
IF(quantidade <= 20, 'Baixo', 'Alto') as Estoque  
  
from produtos;
```

Teste de valores

```
select Valor,  
  
IF(Valor <> 10.00, 'Diferente', 'Igual') AS classificacao  
  
from produtos;
```

Usando função CASE para validar os níveis de Estoque.

```
SELECT  
  
CASE  
  
WHEN Quantidade <= 4 THEN 'Baixo Estoque'  
  
WHEN Quantidade <= 15 THEN 'Estoque Bom'  
  
when Quantidade <= 30 THEN 'Estoque Médio'  
  
ELSE 'Estoque Alto'  
  
END AS Quantidade,  
  
COUNT(*) Quantidade  
  
FROM produtos  
  
GROUP BY Quantidade  
  
ORDER BY Quantidade;
```

Analisar Cliente com compras acima de 10

```
select Nome_Cliente, Nome, Quantidade,  
  
IF(quantidade >= 10, 'Alto', 'Baixo') as Compras_Clientes  
  
from produtos join cadastro on cadastro.id_cad = produtos.id_cad  
  
order by Compras_clientes
```

Trazer Quantidade de Cliente com Vendas Boas e Ruins

**SELECT**

**CASE**

**WHEN Quantidade <= 10 THEN 'Cliente com Poucas Vendas'**

**ELSE 'Clientes com Vendas Boas'**

**END AS Compras\_Clientes,**

**COUNT(\*) Nome\_Cliente**

**FROM produtos join cadastro on produtos.id\_cad = cadastro.id\_cad**

**GROUP BY Compras\_Clientes**

**ORDER BY Compras\_Clientes**