

# Risaltte

Artigos de TI, tutoriais e dicas  
[fique atualizado via rss](#)

## Comandos MySQL

Publicado: 12 de dezembro de 2016 em [Banco de Dados](#), [MySQL](#)

Tags: [MySQL](#)

Lista de comandos MySQL:

### Manipular Bancos

Listar bases de dados (bancos):

```
show databases
```

Criar Banco. Exemplo, criando banco chamado locadora:

```
create database Locadora;
```

Acessar banco recém criado no comando anterior (locadora), para manipula-lo:

```
use Locadora;
```

Listar tabelas do banco. **Obs:** esse comando deve ser executado dentro do banco de qual se deseja ver as tabelas:

```
show tables;
```

### Manipular Tabelas

criar tabela clientes com alguns atributos básicos, comando **CREATE**:

```
create table clientes (cli_codigo integer, cli_nome varchar(30), cli_cidade var
```



## Primary Key (Chave Primária)

Criando a mesma tabela do exemplo anterior com chave principal, que faz com que um determinado campo não seja repetido. Normalmente é criado quando se cria a tabela. **O campo que será a primary key nunca deverá ficar vazio (not null).**

Criar tabela clientes, especificando que os atributos cli\_codigo e cli\_nome não possam ficar vazios e estabelecendo o atributo cli\_codigo como chave primaria:

```
create table clientes (cli_codigo integer not null, cli_nome varchar(30) not nu
```



## Describe

Permite visualizar a estrutura das tabelas

```
describe clientes;
```

## Alter

Permite inserir/eliminar atributos nas tabelas já existentes.

Adicionar um novo atributo (e-mail) na tabela clientes:

```
alter table clientes add cli_email varchar(30);
```

Excluir o atributo (e-mail) na tabela clientes:

```
alter table clientes drop cli_email;
```

Modificar a estrutura de um atributo:

```
alter table clientes modify cli_sexo varchar(10);
```

## UPDATE

Altera os dados de uma tabela.

Modificar o valor do atributo cli\_sexo para todas as tuplas:

```
update clientes set cli_sexo = 'M';
```

Modificar o valor do atributo cli\_sexo onde o atributo cli\_codigo for = 2:

```
update clientes set cli_sexo = 'F' where cli_codigo = 2;
```

Modificando dois atributos

```
update clientes set cli_cidade = 'Bodoquena', cli_nome = 'Risaltte' where cli_cc
```



## Delete

Exclui registros das tabelas

Apaga todos os registros (todas as tuplas) da tabela clientes:

```
delete from clientes;
```

Apagar todos os registros (tuplas) onde o atributo cli\_sexo for = F:

```
delete from clientes where cli_sexo = 'F';
```

Apagar todos os registros (tuplas) onde o atributo cli\_cidade for = Bodoquena:

```
delete from clientes where cli_cidade = 'Bodoquena';
```

Apagar todos os registros (tuplas) onde o atributo cli\_nome iniciar com a letra B:

```
delete from clientes where cli_nome like 'B%';
```

### Truncate

Elimina de forma irreversível todos os dados da tabela, limpando toda a tabela iniciando a inserção de registros do zero novamente. Útil quando se tem atributos com geração de códigos automáticos, pois o delete deixa espaços na tabela.

```
truncate table clientes;
```

### Excluir tabela

Excluir uma tabela e todos seus atributos:

```
drop table clientes;
```

Inserir dados na tabela, comando **INSERT**:

```
insert into clientes (cli_codigo, cli_nome, cli_cidade, cli_sexo) values (1, 'Ra
```

ou

```
insert into clientes values (1, 'Rafael Risalte', 'Bodoquena', 'M');
```

### Visualizar, pesquisar, consultar, selecionar – Comando SELECT:

Visualizar Atributos específicos da tabela clientes:

```
select cli_nome, cli_cidade from clientes clientes;
```

Visualizar todos os Atributos da tabela clientes:

```
select * from clientes;
```

### Clausule Order By – Ordenação

Muda a ordem de apresentação do resultado da pesquisa, possibilitando colocar em ordem ascendente e descendente. Exemplos:

Ordenar pelo atributo cli\_nome (em ordem alfabética ascendente):

```
select * from clientes order by cli_nome asc;
```

ou

```
select * from clientes order by cli_nome;
```

Ordenar pelo atributo cli\_nome (em ordem alfabética descendente):

```
select * from clientes order by cli_nome desc;
```

**Obs:** se **asc** ou **desc** não for especificado a ordenação padrão ascendente (asc) será usada.

Usando duas ordenações. Por exemplo, ordenar por cidade e depois por nomes em ordem alfabética:

```
select * from clientes order by cli_cidade, cli_nome;
```

### Clausula where

Possibilita restringir os dados (tuplas) que são listados, efetuando condição para que os mesmos apareçam na pesquisa.

Operadores lógicos:

= igual a

> maior que

>= maior que ou igual a

< menor que

<= menor que ou igual a diferente Exemplos: Exibir tuplas onde o cli\_codigo seja >= 2:


```
select * from clientes where cli_codigo >= 2;
```

Exibir tuplas onde o cli\_codigo seja >= 2 e <= 5:

```
select * from clientes where cli_codigo >= 2 and cli_codigo <= 5;
```


Exibir tuplas onde o cli\_codigo seja >= 2 e <= 5 e a cidade seja Bodoquena:

```
select * from clientes where cli_codigo >= 2 and cli_codigo <= 5 and cli_cidade
```




Exibir tuplas onde o cli\_codigo seja >= 2 e <= 5 e a cidade seja Bodoquena e o sexo seja Feminino:

```
select * from clientes where cli_codigo >= 2 and cli_codigo <= 5 and cli_cidade
```



Exibir tuplas onde o cli\_codigo seja >= 2 e <= 5 e a cidade seja Bodoquena e o sexo seja Feminino ordenando por nome (cli\_nome):

```
select * from clientes where cli_codigo >= 2 and cli_codigo <= 5 and cli_cidade
```



## Like

Lista tuplas que contenham uma determina string. Exemplos:

Listar todas as tuplas que iniciem com a letra B no atributo cli\_nome:

```
select * from clientes where cli_nome like 'B%';
```

Listar todas as tuplas que terminem com a string Risalte no atributo cli\_nome:

```
select * from clientes where cli_nome like '%Risalte';
```

Listar todas as tuplas que contenham a string Risa em qualquer lugar no atributo cli\_nome:

```
select * from clientes where cli_nome like '%Risa%';
```

### **Between and**

Lista tuplas entre dois valores. Exemplos:

Exibir tuplas onde o cli\_codigo esteja entre 2 e 5:

```
select * from clientes where cli_codigo between 2 and 5;
```

### **In**

Lista de valores pré definidos. Exemplos:

Exibir tuplas onde o cli\_codigo contenha os valores 3 ou 4 ou 7:

```
select * from clientes where cli_codigo in(3,4,7);
```

é a mesma coisa que:

```
select * from clientes where cli_codigo = 3 or cli_codigo = 4 or cli_codigo = 7
```



### **Not between**

Lista todas as tuplas que não estão entre dois valores. Exemplos:

Exibir tuplas que não estejam entre os valores 2 e 5 (menor que 2 e maior que 5) no atributo cli\_codigo:

```
select * from clientes where cli_codigo not between 2 and 5;
```

é a mesma coisa que:

```
select * from clientes where cli_codigo 5;
```

## Not in

Lista os valores que não sejam os pré definidos. Exemplos:

Exibir tuplas onde o cli\_codigo não contenha os valores 3 e 4 e 7:

```
select * from clientes where cli_codigo not in(3,4,7);
```

é a mesma coisa que:

```
select * from clientes where cli_codigo 3 and cli_codigo 4 and cli_codigo 7;
```



## Manipulando atributos e tuplas na visualização dos dados

Mudar títulos dos atributos no resultado da pesquisa:

```
select cli_codigo 'Codigo do Cliente', cli_nome 'Nome do Cliente', Cli_cidade '
```



ou

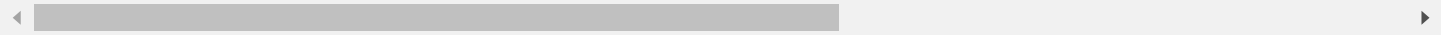
```
select cli_codigo as 'Codigo do Cliente', cli_nome as 'Nome do Cliente', cli_ci
```



Multiplicar o valor do cli\_codigo por 2 no resultado da pesquisa:

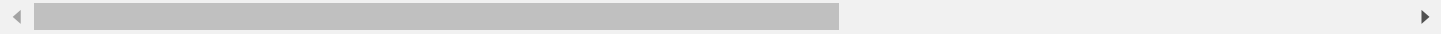


```
select cli_codigo * 2 'Codigo do Cliente', cli_nome 'Nome do Cliente', Cli_cida
```



Soma o valor do cli\_codigo com 2 no resultado da pesquisa:

```
select cli_codigo + 2 'Codigo do Cliente', cli_nome 'Nome do Cliente', Cli_cida
```



## Distinct

Não mostra informações repetidas no resultado da pesquisa, exemplo:

```
select distinct cli_cidade from clientes;
```

## Função AVG

Esta função calcula o valor médio de uma determinada coluna (atributo).

Calcular a média salarial pelo atributo cli\_salario e no resultado das pesquisa exibir o atributo como\_media salarial:

```
select avg (cli_salario) as media_salarial from clientes;
```

sem renomear o atributo no resultado da pesquisa:

```
select avg (cli_salario) from clientes;
```

Calcular a média salarial das mulheres (cli\_sexo = 'F') pelo atributo cli\_salario:

```
select avg (cli_salario) from clientes where cli_sexo = 'F';
```

## Função COUNT

Conta todas as linhas de uma determinada tabela conforme expressão.

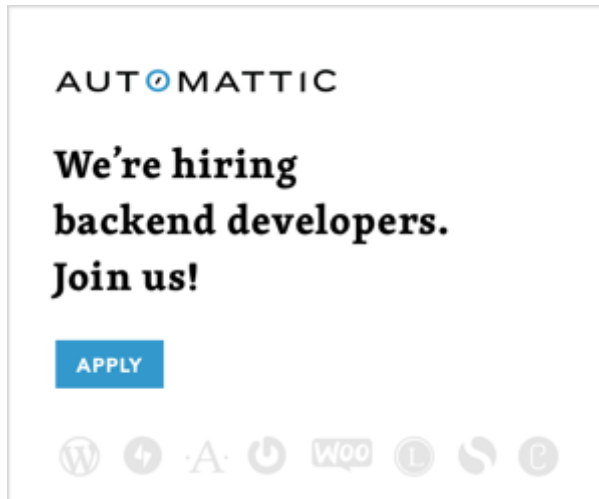
Conta todos os registros na tabela:

```
select count(*) as quantidade_registros from clientes;
```

conta todos os registros do atributo cli\_codigo:

```
select count (cli_codigo) from clientes;
```

Anúncios



DENUNCIAR ESTE ANÚNCIO

Anúncios



DENUNCIAR ESTE ANÚNCIO

Os comentários estão desativados.

[Blog no WordPress.com.](https://risaltte.wordpress.com/2016/12/12/comandos-mysql/)