

Este material pertence a **Carlos A. Majer**, Professor da Unidade Curricular: Aplicações Corporativas da **Universidade Cidade de São Paulo – UNICID**



### Licença de Uso

Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Vedada a Criação de Obras Derivadas 2.5 Brasil. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nd/2.5/br/> ou envie uma carta para Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Este material não pode ser alterado. Para maiores informações envie um e-mail para [cmajer@uol.com.br](mailto:cmajer@uol.com.br) ou [cmajer@ig.com.br](mailto:cmajer@ig.com.br).

# MYSQL

## Introdução

### Pasta para acesso

C:\mysql\bin

Entrada via prompt de comando do DOS, usando o usuário root:-

```
C:\mysql\bin> mysql -u usuário senha
```

Em seguida, o MYSQL exibe seu prompt, pronto para aceitar comandos.


```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.27-community-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>_
```

 *Numa instalação padrão do MySQL, o usuário criado inicialmente é o root e sua senha é vazia.*

Digite o comando desejado, podendo inclusive teclar <ENTER> quantas vezes quiser. Para que o comando seja executado, coloque um ponto e vírgula (;) no final antes de teclar <ENTER>.

 *Alguns comandos não precisam do ponto e vírgula para serem executados.*

### Caixa Alta ou Baixa

O MySQL ignora a caixa dos comandos. Entretanto, tome cuidado pois se você criar tabelas e campos em letras maiúsculas, no **MySQL**, você terá que utilizar a caixa (tamanho de letra) correta, quando estiver trabalhando com o **PHP**, uma vez que o **PHP** não os ignora.

# Manipulando Bancos de Dados

## Algumas tabelas existentes

O Mysql, por conta de sua instalação, já vem com algumas tabelas criadas:-

Mysql: Privilégio de acesso de usuários. Não elimine esta tabela.

Test: Criada como espaço de testes

## Exibindo os bancos de dados existentes

```
SHOW DATABASES;
```

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| sistema |
| test |
+-----+
4 rows in set (0.08 sec)
```

## Abrindo (conectando) a um banco de dados

```
USE [nome do banco de dados];
```

Exemplo:-

```
USE sistema;
```

Mensagem do MySQL:-

```
database changed
```

## Criando um banco de dados

Cria-se uma vez e seleciona-se sempre que precisar trabalhar com o mesmo.

```
CREATE DATABASE [Nome do Banco de Dados];
```

Exemplo:-

```
CREATE DATABASE sistema;
```

Mensagem do MySQL:-

```
Query OK, 1 row affected (0.00 sec)
```

Para começar a trabalhar com este banco de dados, você deve selecioná-lo:-

```
USE sistema;
```

Mensagem do MySQL:-

```
Database changed;
```

## Eliminando Banco de Dados

Para se eliminar um banco de dados, deve-se utilizar o seguinte comando:-

```
DROP [Nome do Banco de Dados]
```

Exemplo:-

```
DROP sistema;
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.00 sec)
```

## Manipulando Tabelas

### Exibindo tabelas do banco de dados aberto (conectado):-

```
SHOW TABLES;
```

Mensagem do MySQL:-

```
+-----+
| Tables_in_sistema |
+-----+
| clientes          |
| produtos          |
+-----+
2 rows in set (0.00 sec)
```

### Exibindo os campos de uma tabela

```
SHOW COLUMNS IN [Nome da Tabela];
```

OU

```
DESCRIBE [Nome da Tabela];
```

Exemplo:-

```
SHOW COLUMNS IN produtos;
```

OU

```
DESCRIBE produtos;
```

Mensagem do MySQL:-

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| codigo     | varchar(10)   | YES  |     | NULL    |       |
| descricao  | varchar(20)   | YES  |     | NULL    |       |
| ativo      | char(1)       | YES  |     | NULL    |       |
| cadastro   | date          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

### Criando tabelas

```
CREATE TABLE [Nome da Tabela] ([Nome do Campo 1] [Tipo do Campo1],
[ Nome do Campo n] [Tipo do Campo n]);
```

Exemplo:-

```
CREATE TABLE Estados
(uf char(2),
regiao VARCHAR(5)
);
```

Mensagem do MySQL:-

Query OK, 0 rows affected (0.11 sec)

### ***Criando tabelas com Chaves Primárias***

```
CREATE TABLE [Nome da Tabela]
([Nome do Campo] [Tipo do Campo] [auto_increment], [Nome do Campo
2] [Tipo do Campo 2],
PRIMARY KEY ([Nome do Campo n]) )
```

Exemplo:-

```
CREATE TABLE clientes
(idCliente INT(10) AUTO_INCREMENT,
nome VARCHAR(50),
cadastro DATE,
nascimento DATE,
PRIMARY KEY (idCliente)
);
```

## **Tipos de Dados**

Na criação das colunas das tabelas, o desenvolvedor deve escolher o tipo de coluna que mais se adequa ao tipo de informação que ele está buscando armazenar. Dentre os tipos que o MySQL têm, podemos citar:-

### ***Inteiro (Integer)***

Este é o tipo de dado utilizado para armazenamento de números. Os números a serem armazenados podem ser positivos ou negativos.

Exemplo:-

-1259, 12, 0, 4, 295, 15344

### ***Caracteres (Char, Varchar)***

Este tipo é utilizado para armazenamento de informações do tipo texto.

Quando utilizado o tipo char, define-se um tamanho que pode ser de 1 até 255 bytes. Independente do tamanho escolhido, qualquer informação que for armazenada fará com que os demais caracteres sejam preenchidos com espaços.

Exemplo:-

Ao gravar o texto "Ola", o tamanho ocupado numa coluna criada originalmente como char(50) será de cinquenta caracteres, sendo "Ola" os seus três primeiros e o restante com espaços.

O tipo **varchar**, por outro lado, permite a gravação de cadeias de caracteres sem gravar o espaço restante. Neste mesmo exemplo, se a coluna tivesse sido criada como varchar(50), o conteúdo a ser gravado seria apenas as três letras (Ola).

### ***Ponto Flutuante (Float)***

Permite a gravação de valores com casas decimais.

Exemplo:-

57.25, 0.577, 23.556

## **Data e Hora**

DATE → Armazena data no formato YYYY-MM-DD

DATETIME → Armazena data e hora no formato YYYY-MM-DD HH:MM:SS

YEAR(2) → Armazena ano no formato de 2 dígitos.

YEAR(4) → Armazena ano no formato de 4 dígitos.

## **Blob**

Tipo de dado binário, o blob pode guardar desde textos até informações advindas de arquivos que foram carregados (processo de upload).

Exemplo:-

```
CREATE TABLE Produtos
(codigo int(10),
titulo VARCHAR(20),
ativo CHAR(1),
cadastro DATE,
especificacao BLOB);
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.11 sec)
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.06 sec)
```

## **Criando tabelas com colunas de valores únicos**

```
CREATE TABLE [Nome da Tabela]
([Nome do Campo] [Tipo do Campo] UNIQUE, [Nome do Campo 2] [Tipo
do Campo 2]);
```

Exemplo:-

```
CREATE TABLE Categorias (codigo VARCHAR(10) unique);
```

## **Eliminando Tabelas**

```
DROP TABLE [Nome da Tabela];
```

Exemplo:-

```
DROP TABLE clientes;
```

## **Alterando Tabelas**

### **Adicionar uma nova coluna (campo)**

```
ALTER TABLE [Nome da Tabela] ADD COLUMN [Nome do Campo] [Tipo do
Campo];
```

Exemplo:-

```
ALTER TABLE Clientes ADD COLUMN Nascimento date;
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.06 sec)
```

```
Registros: 0 - Duplicados: 0 - Avisos: 0
```

### **Alterando o nome de uma coluna**

```
ALTER TABLE [Nome da Tabela] CHANGE [Nome Atual do Campo] [Novo Nome do Campo] [Tipo do Campo Novo];
```

Exemplo:-

```
ALTER TABLE Clientes CHANGE cadastro criacao date;
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.06 sec)
Registros: 0 - Duplicados: 0 - Avisos: 0
```

Alterando uma coluna para que tenha valores únicos

```
ALTER TABLE [Nome da Tabela] ADD UNIQUE [Nome da Coluna];
```

Exemplo:-

```
ALTER TABLE Estados ADD UNIQUE (UF);
```

### **Alterando o tamanho de uma coluna**

```
ALTER TABLE [Nome da Tabela] MODIFY [Nome da Coluna] [Novo Tamanho da Coluna];
```

Exemplo:-

```
ALTER TABLE Estados MODIFY Regiao varchar(15);
```

### **Eliminando uma coluna (campo)**

```
ALTER TABLE [Nome da Tabela] DROP [Nome do Campo];
```

Exemplo:-

```
ALTER TABLE Clientes DROP Nascimento;
```

Mensagem do MySQL:-

```
Query OK, 0 rows affected (0.06 sec)
Registros: 0 - Duplicados: 0 - Avisos: 0
```

Delete unique from table.alter table [table name] drop index [colmn name];

## **Manipulando os registros de tabelas**

### **Inserindo registros numa tabela**

```
INSERT INTO [tabela]
VALUES ([Valor Campo 1], [ Valor Campo 2], [Valor Campo 3]);
```

Exemplos:-

```
INSERT INTO produtos
VALUES ('BC001', 'Caixa de Cereais', '1', '2007-08-19');
```

Mensagem do MySQL:-

Query OK, 1 row affected (0.06 sec)

```
INSERT INTO produtos
VALUES ('BC002', 'Barra de Cereal', '1', NULL);
```

Mensagem do MySQL:-

Query OK, 1 row affected (0.00 sec)

- ✎ *Você deve inserir os valores dos campos, conforme sua seqüência de criação, ou seja, os valores dos campos dentro dos parênteses, após VALUES, serão gravados (caso haja sucesso) conforme a ordem de criação dos campos na tabela.*
- ✎ *Você pode utilizar o valor NULL para deixar um campo sem informação alguma*

A tentativa de gravação de um registro contendo valores para um número diferente de colunas da tabela irá fazer com que um erro seja gerado:-

```
INSERT INTO produtos VALUES ('BC003');
```

Mensagem do MySQL:-

ERROR 1136 (21S01): Contagem de colunas não confere com a contagem de valores na linha 1

## Exibindo registros de uma tabela

```
SELECT [* | Campo 1, Campo n] FROM [Nome da Tabela]
```

Exemplo:-

```
SELECT * FROM produtos;
```

Mensagem do MySQL:-

```
+-----+-----+-----+-----+
| codigo | descricao          | ativo | cadastro |
+-----+-----+-----+-----+
| BC001  | Caixa de Cereais  | 1     | 2007-08-19 |
| BC002  | Barra de Cereal   | 1     | NULL      |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

O desenvolvedor também restringir o resultado da exibição conforme um filtro lógico.

```
SELECT * FROM produtos WHERE codigo='BC002';
```

Mensagem do MySQL:-

```
+-----+-----+-----+-----+
| codigo | descricao          | ativo | cadastro |
+-----+-----+-----+-----+
| BC002  | Barra de Cereal   | 1     | NULL      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Contando o número de registros de uma tabela

```
SELECT COUNT(*) FROM [Nome da Tabela];
```

Exemplo:-

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

Mensagem do MySQL:-

```
+-----+
| count(*) |
+-----+
|         2 |
+-----+
1 row in set (0.00 sec)
```

## Alterando registros de uma tabela

```
UPDATE [Nome da Tabela] SET [Campo 1] = [Valor], [Campo 2] = [Valor]
```

Exemplo:-

```
UPDATE produtos SET ativo='1';
```

Mensagem do MySQL:-

```
Query OK, 2 rows affected (0.00 sec)
Linhas que combinaram: 2 - Alteradas: 2 - Avisos: 0
```

 *No exemplo acima, todos os registros da tabela são alterados.*

Caso o usuário deseje alterar um conjunto de registros, ele deve utilizar uma expressão lógica.

```
UPDATE [Nome da Tabela] SET [Campo 1] = [Valor], [Campo 2] = [Valor] WHERE [Condição Lógica]
```

Exemplo:-

```
UPDATE produtos SET ativo='0' WHERE CODIGO='BC002';
```

Mensagem do MySQL:-

```
Query OK, 1 row affected (0.00 sec)
Linhas que combinaram: 1 - Alteradas: 1 - Avisos: 0
```

Esta condição lógica pode ser uma expressão complexa:-

```
UPDATE produtos SET ativo='0' WHERE ativo='1' and cadastro>='2007-01-01';
```

Mensagem do MySQL:-

```
Query OK, 1 row affected (0.00 sec)
Linhas que combinaram: 1 - Alteradas: 1 - Avisos: 0
```

## Eliminando registros de uma tabela

O desenvolvedor pode eliminar todos os registros de uma tabela através do seguinte comando:-

```
DELETE FROM [Nome da Tabela] WHERE [Condição];
```

Exemplo:-



```
DELETE FROM produtos;
```

Mensagem do MySQL:-

```
Query OK, 2 rows affected (0.01 sec)
```

Caso deseje eliminar um conjunto de registros, o desenvolvedor deve utilizar-se de uma condição lógica:-

```
DELETE FROM [Nome da Tabela] WHERE [Condição Lógica]
```

Exemplo:-

```
DELETE FROM produtos WHERE codigo='BC002';
```

Mensagem do MySQL:-

```
Query OK, 1 row affected (0.02 sec)
```

## Backup e Restauração de Banco de Dados

Você pode utilizar a ferramenta (programa) MySQLDump para efetuar o backup (cópia de segurança) e a restauração de um banco de dados MySQL.

### Backup de um banco de dados


A sintaxe para se fazer um backup é:-

```
MYSQLDUMP -U [Nome do Usuário] -P[Senha] [Nome do Banco de Dados]  
> [Nome do Arquivo de Backup]
```

Exemplo:-

```
MYSQLDUMP -u root sistema > sistema.sql
```

O arquivo **sistema.sql** será criado com todas as instruções em SQL para criação e alimentação do banco de dados de nome sistema, com todos os seus campos e registros.

 *Nada será exibido na tela, devendo ser exibido apenas o prompt do MS-DOS, uma vez que todo o conteúdo está sendo redirecionado para o arquivo sistema.sql. Caso deseje verificar o resultado na tela, simplesmente tire o texto "> sistema.sql" do comando acima.*

### Restaurando um banco de dados

```
MYSQL -u [Nome do Usuário] -p [Senha] [Nome do Banco de Dados] <  
[Arquivo de Backup]
```

Exemplo:-

```
MYSQL -u root sistema < sistema.sql
```

 *O banco de dados deverá já existir no MySQL*