

CONFIGURAÇÃO DO MARIADB

INDEX

1. [Descobrir onde os dados são gravados](#)
2. [Como alterar a base de dados padrão do MySQL](#)
3. [Como ajustar a segurança do Banco de dados](#)
4. [Como configurar mariadb para que permita acesso remoto](#)
5. [Como criar DataBase no mariadb](#)
6. [Usuários](#)
 1. [Como adicionar usuários](#)
 2. [Como excluir usuários](#)
7. [Variáveis de sistema do servidor mariadb](#)
8. [Como o mariadb lida com letras maiúsculas e minúsculas](#)
9. [Como instalar mariadb](#)
10. [REFERÊNCIAS](#)

CONTEÚDO

1. Descobrir onde os dados são gravados

1. Para saber **onde os dados são gravados no mariadb** usar a sequencia abaixo:

1. Entrar no cliente mysql do mariadb do mariadb.

```
1. $ sudo mysql -u root -p
```

2. Para descobrir a pasta onde os dados são gravados executar o seguinte comando:

```
1. SELECT @@datadir;
```

3. Para visualizar todas as variáveis do mariadb:

```
1. show variables like "%DATA%";
```

4. Para sair do cliente mysql do mariadb:

```
1. quit;
```

2. Como alterar a base de dados padrão do MySQL

1. A pasta de dados padrão do MySQL é: `/var/lib/mysql`.

2. Como mudar a pasta de dados de `"/var/lib/mysql"` para `"/mariadbdatar"`:

1. O mariadb instalado é a ultima versão estável no momento da publicação desta dica:
`ubuntu_server-mate 5.4.0-60-generic`.

2. A etapa abaixo só funciona se a pasta estiver na raiz `/`. Tentei muito fazer funcionar na pasta `/home/mariadbdatar`, mais não funciona.

3. Etapas do processo:

1. Parar a execução do mariadb:

```
1. $ sudo systemctl stop mariadb
```

2. Criar pasta customizada **`/mariadbdatar`**.

```
1. $ sudo mkdir /mariadbdatar
```

3. Informa ao linux que o dono da pasta `/mariadbdatar` é o usuário `mysql:mysql`.

```
1. $ sudo chown -R mysql:mysql /mariadbdatar
```

4. Editar o arquivo **`"/etc/mysql/my.cnf"`** e aponte para o novo diretório de dados:

```
1. $ sudo vi /etc/mysql/my.cnf
```

2. Adicione no final do arquivo **`/etc/mysql/my.cnf`** as linhas abaixo:

```
1. [mysqld]
   datadir=/mariadbdatar
```

5. Copia todos as pastas e arquivos da pasta anterior (exceto links simbólicos) para a nova pasta:

```
1. $ sudo cp -vRp /var/lib/mysql/* /mariadbdatar
```

6. Habilita mariadb:

```
1. $ sudo systemctl enable mariadb
```

7. Inicializa mariadb:

```
1. $ sudo systemctl start mariadb
```

8. Checa se foi inicializado:

```
1. systemctl status mariadb
```

9. Inicializa o linux para que mariadb se ajusta as mudanças.

```
1. reboot
```

3. Caso ocorra mensagem de muitos arquivos aberto, entrar no arquivo abaixo e alterar o padrão:

1. Arquivo `/etc/systemd/system/mysql.service` contém a variável `LimitNOFILE=16384` que determina o número máximo de arquivos abertos.

4.  TOP

3. Como ajustar a segurança do Banco de dados

1. **mysql_secure_installation** é um script de shell disponível em sistemas Unix e permite que você melhore a segurança da instalação do MariaDB das seguintes maneiras:

1. Você pode definir uma senha para contas root.
2. Você pode remover contas root que estão acessíveis de fora do host local.
3. Você pode remover contas de usuários anônimos.
4. Você pode remover o banco de dados de teste, que por padrão pode ser acessado por usuários anônimos.
5. O **mysql_secure_installation** pode ser invocado sem argumentos:

```
1. $ sudo mysql_secure_installation
```

1. As mensagens abaixo foi traduzida das mensagens geradas pelo script *mysql_secure_installation*.

1. NOTA: EXECUTAR TODAS AS PARTES DESTE SCRIPT É RECOMENDADO PARA TODO O MariaDB. SERVIDORES EM USO DE PRODUÇÃO! POR FAVOR, LEIA ATENTAMENTE CADA PASSO.

2. ETAPA PARA ALTERAR A SENHA DO ROOT

1. Para entrar no MariaDB para protegê-lo, precisaremos da senha atual do usuário root. Se você acabou de instalar o MariaDB e ainda não definiu a senha do root, a senha estará em branco, então você deve apenas pressionar Enter aqui.

```
1. Enter current password for root
   (enter for none):
```

2. Definir a senha de root garante que ninguém possa logar no usuário root do MariaDB sem a devida autorização. Se você já tem uma senha de root definida, pode responder com segurança 'n'.

```
1. Change the root password? [Y/n] y
```

```
1. New password:
```

```
2. Re-enter new password:
```

3. ETAPA PARA EXCLUIR OU NÃO O USUÁRIO ANÔNIMO*

1. Por padrão, uma instalação do MariaDB tem um usuário anônimo, permitindo que qualquer pessoa se logue no MariaDB sem ter que ter uma conta de usuário criada para eles. Destina-se apenas a teste e para tornar a instalação um pouco mais suave. Você deve removê-los antes de mudar para um ambiente de produção.

```
1. Remove anonymous users? [Y/n] y
```

4. ETAPA PARA EXCLUIR OU NÃO O BANCO DE DADOS TESTE

1. Por padrão, MariaDB vem com um banco de dados chamado 'teste' que qualquer pessoa pode acessar. Isso também se destina apenas a teste e deve ser removido antes de mudar para um ambiente de produção.

```
1. Remove test database and access to
   it? [Y/n] y
```

5. ETAPA PARA RECARREGAR OU NÃO A TABELA DE PRIVILÉGIOS DO BANCO DE DADOS

1. Recarregar as tabelas de privilégios garantirá que todas as alterações feitas até agora terão efeito imediato.

1. `Reload privilege tables now? [Y/n] y`

2. 

4. Como configurar mariadb para que permita acesso remoto

1. Para que todos os computadores acessem o banco de dados, edite o arquivo `/etc/mysql/my.cnf` e adicione no final do arquivo as linhas:

1. `$ sudo vi /etc/mysql/my.cnf`
//em seguida adicione no final do arquivo `/etc/mysql/my.cnf`
`[mysqld]`
`bind-address=*`

2. Execute o comando abaixo no prompt do sistema para reiniciar o banco de dados mariadb:

1. `$ sudo systemctl restart mariadb`

3. Habilite o usuário root para acessar de qualquer computador.

1. Entrar no cliente mysql do mariadb do mariadb.

1. `mysql -h localhost -u root -p`

2. Dê permissão para que o usuário root acesse fora do localhost.

1. `GRANT ALL ON *.* TO 'root'@'%' IDENTIFIED BY 'password';`

1. Onde: O sinal % significa "qualquer IP", ou seja, indica que o usuário pode acessar o banco de dados em qualquer máquina da rede. Para habilitar o acesso apenas para um único IP, substitua o sinal % pelo IP desejado, assim somente a máquina com o IP indicado poderá conectar o servidor.

3. Para sair do cliente mysql:

```
1. quit;
```

4.  TOP

5. Como criar DataBase no mariadb

1. Entrar no cliente mysql do mariadb do mariadb.

```
1. $ sudo mysql -u root -p
```

2. Cria o banco de dados nomeDataBase:

```
1. CREATE DATABASE nomeDataBase;
```

3. Para visualizar todos os DATABASES:

```
1. SHOW DATABASES;
```

4. Para selecionar o banco de dados criado (nomeDataBase) executar comando:

```
1. USE nomeDataBase;
```

5. Para visualizar as tabela do database selecionado:

```
1. show tables
```

6. SHOW CHARACTER SET

1. Sintaxe:

```
1. SHOW CHARACTER SET [LIKE 'pattern' | WHERE expr]
```

2. Exemplo:

1. Para checar a lista de opções de tipo de caractere e default collation em latin:

```
1. show character set like 'latin%'
```

7. Definir conjuntos de caracteres e agrupamentos:

1. Exemplo:

1. Para não distinguir letras minúsculas e maiúsculas e nem caracteres acentuados e com cedilha, ou seja, o registro que contém a palavra "Intuição" será retornado quando houver uma procura pela palavra "intúicao"

```
1. ALTER DATABASE nomeDataBase CHARACTER SET =  
'latin1' COLLATE = 'latin1_swedish_ci';
```

8. Tabela Esquema de Informação SCHEMATA:

1. Exemplo:

```
1. SELECT * FROM INFORMATION_SCHEMA.SCHEMATA\G
```

9. Para sair do cliente mysql:

```
1. quit;
```

2. Obs: O nome do banco de dados usa o mesmo estilo de nomes da linguagem c++, ou seja maiúsculas e minúsculas são diferentes, porém é possível mudar o padrão mudando a variável

lower_case_table_names=1 em /etc/mysql/my.cnf .

3.  [TOP](#)

6. Usuários

6.1 Adicionando usuários

6.1.1 Como adicionar usuários ao banco de dados

1. Entrar no cliente mysql do mariadb do mariadb:

```
1. $ sudo mysql -h localhost -u root -p
```

2. Para GRANT ALL conceder privilégios a um user, permitindo que o usuário controle total sobre um determinado database se tiver no localhost, use a seguinte sintaxe:

1. Sintaxe:

1. GRANT ALL PRIVILEGES ON database_name.* TO 'user_name'@'localhost';

2. Exemplo:

1. GRANT ALL PRIVILEGES ON Db_BlogPsspAppBr.* TO
'Use_BlogPsspAppBr'@'localhost';

3. Para criar o usuário "User_BlogPsspAppBr" com permissão de acesso a qualquer computador:

1. CREATE USER 'User_BlogPsspAppBr'@'%' IDENTIFIED BY 'senha';

4. Para criar o usuário "User_BlogPsspAppBr" com privilégio de administrador, porém só pode acessar o banco de dados pelo IP:"192.168.15.5":

1. GRANT ALL PRIVILEGES ON *.* TO
'BlogPsspAppBr'@'192.168.15.5' IDENTIFIED BY 'password'
WITH GRANT OPTION;

2. FLUSH PRIVILEGES;

5. Para visualizar todos os usuários cadastrados:

1. SELECT User FROM mysql.user;

6. Para sair do cliente mysql:

1. quit;

2.  TOP

6.1.2 Como saber os privilégios dos usuários

1. Entrar no cliente mysql do mariadb:

1. \$ sudo mysql -h localhost -u root -p

2. SELECT User, Host FROM mysql.user WHERE Host <> 'localhost';

3. Para sair do cliente mysql:

```
1. quit;
```

2.  TOP

6.2 - Excluindo usuários

1. O DROP USER extrato remove uma ou mais contas MariaDB. Ele remove linhas de privilégios para a conta de todas as tabelas de permissões. Para usar esta instrução, você deve ter o privilégio global CREATE USER ou o privilégio DELETE para o banco de dados mysql. Cada conta é nomeada usando o mesmo formato do CREATE USER extrato; por exemplo 'jeffrey'@'localhost',. Se você especificar apenas a parte do nome de usuário do nome da conta, uma parte do nome do host de '%'será usada. Para obter informações adicionais sobre como especificar nomes de contas, consulte CREATE USER.Observe que, se você especificar uma conta que está conectada no momento, ela não será excluída até que a conexão seja fechada. A conexão não será fechada automaticamente.

1. Exemplo:

1. Entrar no cliente mysql do mariadb:

```
1. $ sudo mysql -h localhost -u root -p
```

2. Apaga o usuário:

```
1. DROP USER Nome_do_usuario;
```

3. Para sair do cliente mysql:

```
2. quit;
```

2.  TOP

7. Variáveis de sistema do servidor mariadb

1. Como visualizar todas as variáveis de servidor mariadb:

```
1. $ sudo mysqld --verbose --help
```

2. Para ver os valores que um servidor usa com base apenas em seus padrões compilados, ignorando as configurações em qualquer arquivo de opção, use este comando:

```
1. $ sudo mysqld --no-defaults --verbose --help
```

3. Opções de inicialização para o mysqld.

1. Todas as opções de mysqld.

2. Os parâmetros do mariadb-server está no arquivo `/etc/mysql/my.cnf`.

1. Exemplos de configurações customizadas:

1. [mysqld]

1. `lower_case_table_names=1` //[Os nomes de variáveis e tabelas usam o padrão windows]

2. `datadir =/mysql` //A pasta com os dados

4.  TOP

8. Como o mariadb lida com letras maiúsculas e minúsculas

1. Sensibilidade do Identificador de Caso:

1. No MySQL, os bancos de dados correspondem a diretórios dentro do diretório de dados. Cada tabela em um banco de dados corresponde a pelo menos um arquivo no diretório do banco de dados (e possivelmente mais, dependendo do mecanismo de armazenamento). Os gatilhos também correspondem a arquivos. Conseqüentemente, a distinção entre maiúsculas e minúsculas do sistema operacional subjacente desempenha um papel na distinção entre maiúsculas e minúsculas do banco de dados, tabela e nomes de gatilhos. Isso significa que esses nomes não diferenciam maiúsculas de minúsculas no Windows, mas na maioria das variedades do Unix. Uma exceção notável é o macOS, que é baseado em Unix, mas usa um tipo de sistema de arquivos padrão (HFS +) que não faz distinção entre maiúsculas e minúsculas. No entanto, o macOS também oferece suporte a volumes UFS, que diferenciam maiúsculas de minúsculas, assim como em qualquer Unix. Veja Seção 1.7.1, "Extensões MySQL para SQL padrão". A `lower_case_table_names` variável do sistema também afeta como o servidor lida com a diferenciação de maiúsculas e minúsculas do identificador, conforme descrito posteriormente nesta seção.

2. O modo como os nomes das tabelas e bancos de dados são armazenados no disco e usados no MySQL é afetado pela `lower_case_table_names` variável do sistema. `lower_case_table_names` pode assumir os valores mostrados na tabela a seguir. Esta variável não afeta a diferenciação de maiúsculas e minúsculas dos identificadores de gatilho. No Unix, o valor padrão `lower_case_table_names` é 0. No Windows, o valor padrão é 1. No macOS, o valor padrão é 2. `lower_case_table_names` só pode ser configurado ao inicializar o servidor. `lower_case_table_names` é proibido alterar a configuração após a inicialização do servidor.

1. Valores da variável `lower_case_table_names`.

1. **0 (zero)**

1. Os nomes de tabelas e bancos de dados são armazenados no disco usando as letras maiúsculas especificadas na instrução **CREATE TABLE** ou **CREATE DATABASE**. As comparações de nomes diferenciam maiúsculas de minúsculas.

Você não deve definir esta variável como 0 se estiver executando o MySQL em um sistema que tenha nomes de arquivo que não diferenciam maiúsculas de minúsculas (como Windows ou macOS). Se você forçar essa variável para 0 -- **lower-case-table-names = 0** em um sistema de arquivos que não faz MyISAM distinção entre maiúsculas e minúsculas e acessar nomes de tabelas usando diferentes maiúsculas e minúsculas, o índice poderá ser corrompido.

2. **1 (um)** Obs: Penso que o ideal é não diferenciar maiúsculas de minúsculas.

1. Os nomes das tabelas são armazenados em letras minúsculas no disco e as comparações de nomes não diferenciam maiúsculas de minúsculas. O MySQL converte todos os nomes de tabelas em letras minúsculas no armazenamento e na pesquisa. Esse comportamento também se aplica a nomes de banco de dados e aliases de tabela.

3. **2 (dois)**

1. Os nomes de tabelas e bancos de dados são armazenados em disco usando letras maiúsculas especificadas na instrução **CREATE TABLE** ou **CREATE DATABASE**, mas o MySQL os converte em minúsculas na pesquisa. As comparações de nomes não diferenciam maiúsculas de minúsculas. Isso funciona apenas em sistemas de arquivos que não fazem distinção entre maiúsculas e minúsculas! InnoDB nomes de tabelas e nomes de visualizações são armazenados em letras minúsculas, assim como para **lower_case_table_names=1**.

3. 

REFERÊNCIAS

1. [MySQL 8.0 Reference Manual](#)
2. [Como-alterar-a-base-de-dados-padrao-do-MySQL](#)
3. [mysql_secure_installation](#)
4. [Como ajustar a segurança do Banco de dados](#)
5. [using-mysqld_install_db](#)
6. [Variáveis de sistema do servidor](#)
7. 