

GERENCIAMENTO DE USUÁRIOS VIA CMD

CRIANDO USUÁRIO NO MYSQL

```
mysql> CREATE USER 'novo-usuário'@'localhost' IDENTIFIED BY 'senha';
```

NOTA

Vamos especificar o host do usuário como localhost e não o endereço IP do servidor. localhost é um nome de host que significa “este computador”, e o MySQL trata esse nome de host em particular de maneira especial: quando um usuário com aquele host se loga no MySQL, ele tentará se conectar ao servidor local usando um arquivo de soquete Unix. Assim, localhost é normalmente usado quando você planeja se conectar por SSH em seu servidor ou quando você está executando o cliente local mysql para se conectar ao servidor MySQL local.

PERMISSÕES PARA O NOVO USUÁRIO

Neste ponto o **novo-usuário** não tem nenhuma permissão para realizar algo no banco de dados, inclusive nem logar, mesmo com a **senha** criada.

Para dar permissões totais ao novo usuário

```
mysql> GRANT ALL PRIVILEGES ON * . * TO 'novo-usuário'@'localhost';
```

NOTA

Os asteriscos neste comando referem-se ao banco de dados e à tabela (respectivamente) que eles podem acessar - este comando específico permite ao usuário ler, editar, executar e realizar todas as tarefas em todos os bancos de dados e tabelas. Observe que estamos concedendo ao novo usuário acesso root total a tudo em nosso banco de dados. Embora isso seja útil para explicar alguns conceitos do MySQL, pode ser impraticável para a maioria dos casos de uso e pode colocar a segurança do seu banco de dados em alto risco.

Após finalizar as novas permissões é necessário recarregar todas as permissões.

```
mysql> FLUSH PRIVILEGES;
```

Para dar permissões diferentes aos usuários

Tipos comuns de possíveis permissões dadas aos usuários

- ALL PRIVILEGES - isso permitiria a um usuário MySQL acesso total a um banco de dados designado (ou se nenhum banco de dados for selecionado, acesso global em todo o sistema)
- CREATE - permite que os usuários criem tabelas ou bancos de dados
- DROP - permite que os usuários excluam tabelas ou bancos de dados
- DELETE - permite excluir linhas das tabelas
- INSERT - permite inserir linhas nas tabelas
- SELECT - permite que eles usem o comando SELECT para leitura
- UPDATE - permite que eles atualizem as linhas da tabela

GERENCIAMENTO DE USUÁRIOS VIA CMD

- GRANT OPTION - permite conceder ou remover privilégios de outros usuários

Para especificar as permissões usar o comando

```
mysql> GRANT tipo-de-permissão ON nome-bd.nome-da-tabela TO 'novo-usuário'@'localhost';
```

Para alterar modo de autenticação do usuário

```
mysql> ALTER USER 'usuário'@'localhost' IDENTIFIED WITH mysql_native_password BY 'senha';  
mysql> FLUSH PRIVILEGES;
```

NOTA

mysql_native_password é normalmente o método de autenticação padrão. Com WITH você pode decidir qual método usar. Por exemplo, se você usar GRANT USAGE ON *.* TO root @ localhost IDENTIFIED WITH socket_auth, você está configurando esse usuário para usar autenticação de soquete unix. MariaDB usa uma sintaxe ligeiramente diferente: VIA unix_socket. A execução desses comandos resulta principalmente em uma atualização da tabela mysql.user.

Para remover a permissão de um usuário

```
mysql> REVOKE tipo-de-permissão ON nome-bd.nome-da-tabela FROM 'novo-usuário'@'localhost';
```

Para ver as permissões de um usuário

```
mysql> SHOW GRANTS FOR 'novo-usuário'@'localhost';
```

Para excluir o usuário

```
mysql> DROP USER 'novo-usuário'@'localhost';
```

Para visualizar todos os usuários criados

```
mysql> SELECT user FROM mysql.user;
```