

Linguagem de Banco de Dados

SQL

Structure Query Language

Laércio Silva

Indsilva@hotmail.com

Banco de Dados Relacional

O que é um banco de dados Relacional

- Foi originalmente desenvolvido em 1970 pelo Dr. E.F. Codd (*Edgar Frank Codd*);
- Esboçou a teoria de banco de dados relacionais em seu artigo intitulado ["A Relational Model of Data for Large Shared Data Banks"](#) (Um modelo de dados relacional para grandes bancos de dados compartilhados);
- Publicado na Communications of the ACM (Association for Computing Machinery), vol. 13, nº 6, julho de 1970;

Banco de Dados Relacional

O que é um banco de dados Relacional

- É uma coleção de informações relacionadas, organizadas em *tabelas*;
- Cada tabela armazena dados em *linhas*;
- Os dados são organizados em *colunas*;
- As tabelas são armazenadas em esquemas de banco de dados, que são áreas onde os usuários podem armazenar suas próprias tabelas;
- Um usuário pode conceder permissões a outros usuários para que eles possam acessar suas tabelas.

Banco de Dados Relacional

- Todo o banco de dados precisa ter algum modo de inserir e extrair dados, de preferência usando uma linguagem comum, entendida por todos os banco de dados;
- Os sistemas de gerenciamento de banco de dados implementam um linguagem conhecida como ***Structure Query Language ou SQL;***
- ***SQL*** – permite recuperar, adicionar, modificar e excluir informações em um banco de dados.

Banco de Dados Relacional

- Um ponto importante a ser lembrado é que as informações que compõem um banco de dados são diferentes do sistema usado para acessar um essas informações;
- O software usado para acessar um banco de dados é conhecido como *sistema de gerenciamento de banco de dados*.
 - *Exemplo de SGBD: Oracle, SQL Server, DB2, MySQL, dentre outros.*

Apresentando a Linguagem SQL

- A linguagem SQL (*Structure Query Language*) é a linguagem padrão projetada para acessar banco de dados relacionais;
- A linguagem SQL é baseada no trabalho pioneiro do Dr. E.F. Codd;
- Sua primeira implementação foi desenvolvida pela IBM em meados dos anos 1970, dentro do projeto de pesquisa conhecido como System R.
- A linguagem SQL está totalmente padronizada e é reconhecida pelo *American National Standards Institute (ANSI)*.

Grupos de comandos SQL

Os comandos do SQL são classificados em três grupos, de acordo com suas principais funções:

- **DDL (Data Definition Language):** são todos aqueles comandos usados para criar e alterar tabelas que compõem o banco de dados, ou seja, os comandos que definem a estrutura dos dados;
- **DML (Data Manipulation Language):** pertencem a este grupo todos os comandos usados para extrair informações das tabelas, ou seja, para manipular os dados existentes;
- **DCL (Data Control Language):** trata-se de um conjunto de comandos usado em sistemas multiusuário para definir os privilégios de acesso aos dados a cada usuário. Os comandos de controle de acesso aos dados são usados para implementar segurança e privacidade em bancos de dados.

Apresentando a Linguagem SQL

- **Instruções DDL (Data Definition Language)** definem as estruturas de dados, como as tabelas, que compõem um banco de dados. Existem cinco tipos básicos de instruções DDL:
 - **CREATE** cria uma estrutura de banco de dados. Exemplo, *CREATE DATABASE* e usado para criar banco de dados, *CREATE TABLE* é usada para criar uma tabela; outro exemplo *CREATE USER*, usada para criar um usuário do banco de dados.
 - **ALTER** modifica uma estrutura de banco de dados. Exemplo, *ALTER TABLE* é usada para modificar uma tabela.
 - **DROP** remove uma estrutura de banco de dados. Exemplo, *DROP TABLE* é usada para remover uma tabela;

Apresentando a Linguagem SQL

- A linguagem SQL usa uma sintaxe simples, fácil de aprender e utilizar;
- Existem cinco tipos de instruções SQL:
 - **Instruções de consulta:** recuperam linhas armazenadas nas tabelas do banco de dados. **SELECT**
 - **Instruções DML(Data Manipulation Language)** modificam o conteúdo das tabelas. Existem três instruções DML:
 - **INSERT** adiciona linhas em uma tabela.
 - **UPDATE** altera linhas.
 - **DELETE** remove linhas

Apresentando a Linguagem SQL

- **RENAME** muda o nome de uma tabela;
- **TRUNCATE** exclui todas as linhas de uma tabela.
- **Instruções TC (Transaction Control)** registram permanentemente as alterações feitas em linhas ou desfazem essas alterações. Existem três instruções TC:
 - **COMMIT** registra permanente as alterações feitas em linhas;
 - **ROLLBACK** desfaz as alterações feitas em linhas;
 - **SAVEPOINT** define um “ponto de salvamento” no qual pode-se reverter alterações.

Apresentando a Linguagem SQL

- **Instruções DCL (Data Control Language)** alteram as permissões nas estruturas de banco de dados. Existem duas instruções DCL:
 - **GRANT** concede a outro usuário acesso às estruturas de seu banco de dados;
 - **REVOKE** impede que outro usuário acesse as estruturas de seu banco de dados.
- Existem muitas maneiras de executar instruções SQL e obter resultados do banco de dados, vamos utilizar o **MySQL / MariaDB**.

Conexão com MySQL

- **MySQL Installer** não instalar somente para conhecimento:

<http://dev.mysql.com/downloads/windows/installer/>

- **MySQL Workbench** - não instalar somente para conhecimento :

<http://dev.mysql.com/downloads/tools/workbench/>

- **Vamos utilizar em aula esse SQBD - Xampp installer:**

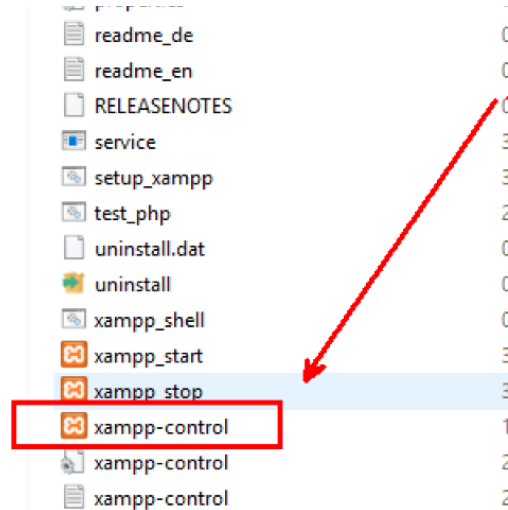
https://www.apachefriends.org/pt_br/index.html

Conectando ao MySQL

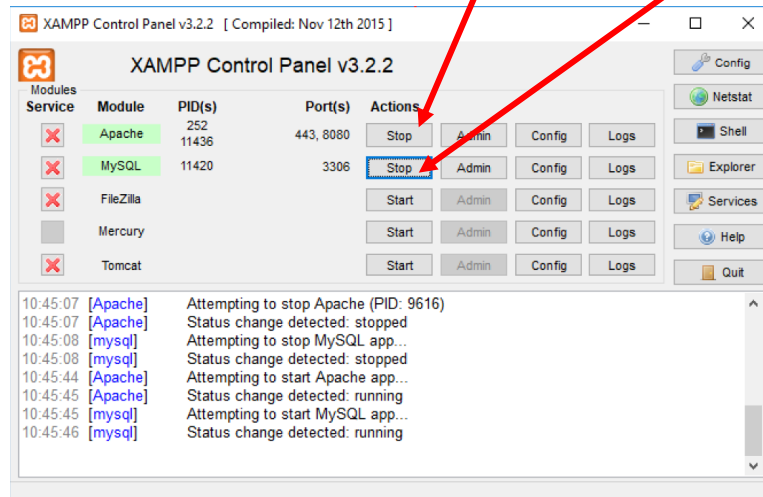
Carregar o servidor em:

cd xampp/

E clicar no arquivo

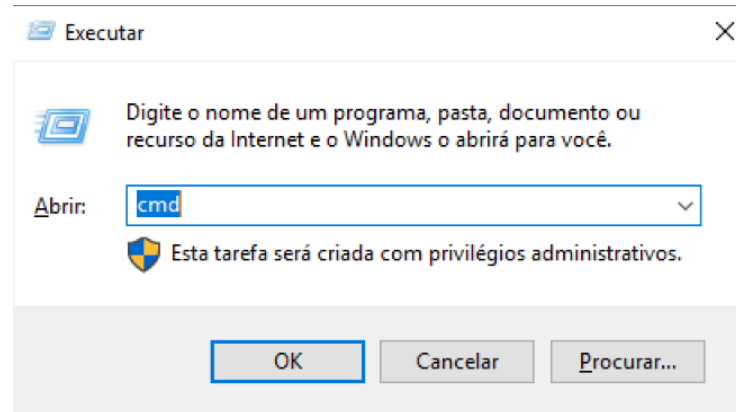


E acionar o serviço de "Apache" e "MYSQL"



Acessando o MySQL

Windows + R



Acessar o diretório c:/xampp/mysql/bin pelo prompt

Usar o comando:

cd xampp/mysql/bin

```
C:\xampp>cd mysql  
C:\xampp\mysql>cd bin  
C:\xampp\mysql\bin>
```

Conectando ao MySQL

Precisamos utilizar um comando para acessar o prompt do MySQL.

Ao instalarmos o MySQL é obrigatório criar um usuário e senha para o acesso dos Banco de Dados.

Por padrão, o usuário é **root** e a senha é vazia.

Estas informações (usuário e senha) são necessários para este passo.

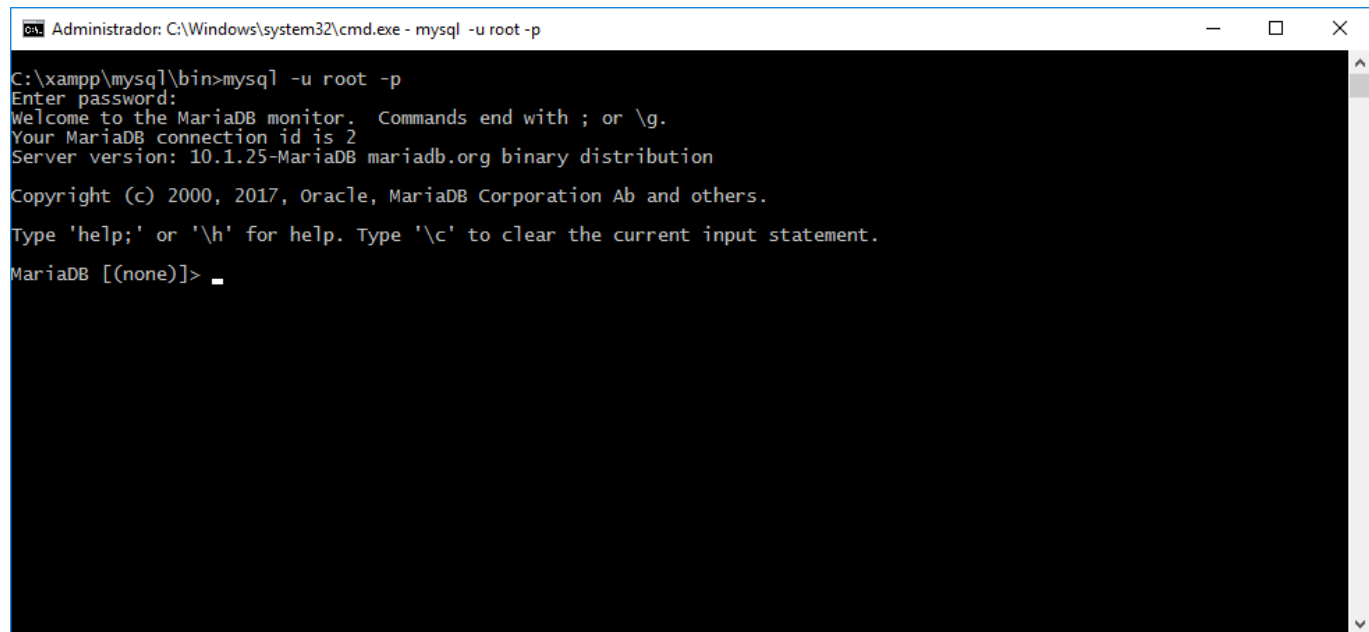
Conectando ao MySQL

O comando para acessarmos o MySQL é:

`mysql -u usuario -p senha`

Em nosso caso ficando:

`mysql -u root -p`



```
Administrador: C:\Windows\system32\cmd.exe - mysql -u root -p

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.25-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```


Conectando ao MySQL

Descrição dos comandos:

`mysql [-h nome_maquina] [-u nome_usuario] [-p sua_senha]`

usuario -u, --user=name

host -h, --host=name

senha -p, --password=name

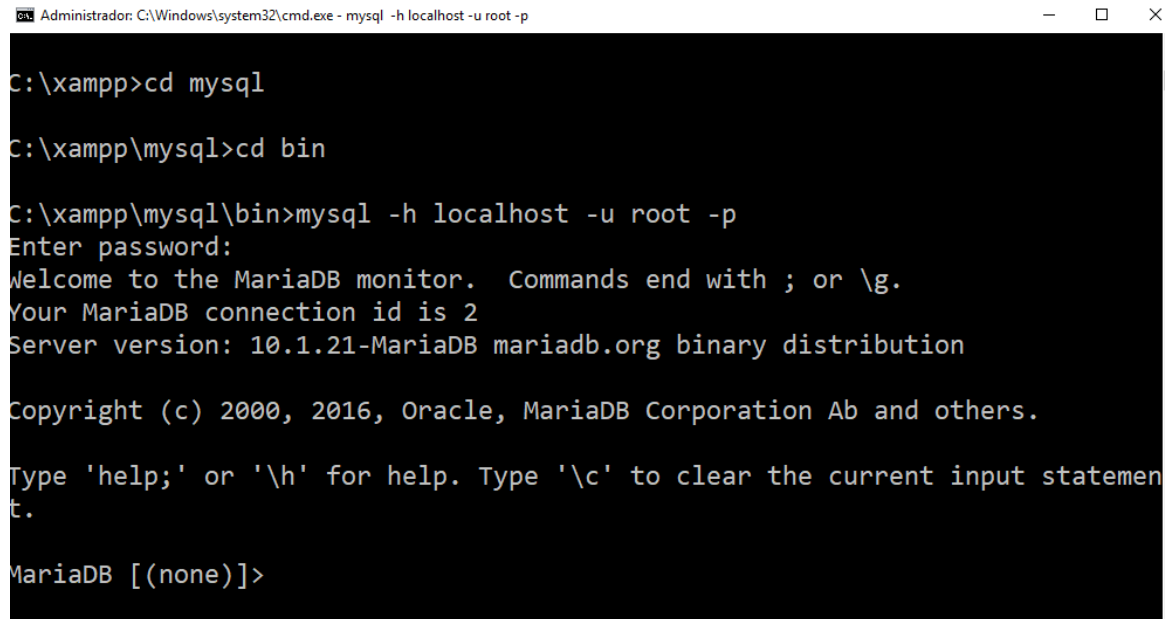
database -D, --database=name

porta -P, --port=3306

versão -V, --version

Exemplo:

`mysql -h localhost -u root -p`



```
Administrador: C:\Windows\system32\cmd.exe - mysql -h localhost -u root -p

C:\xampp>cd mysql
C:\xampp\mysql>cd bin
C:\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

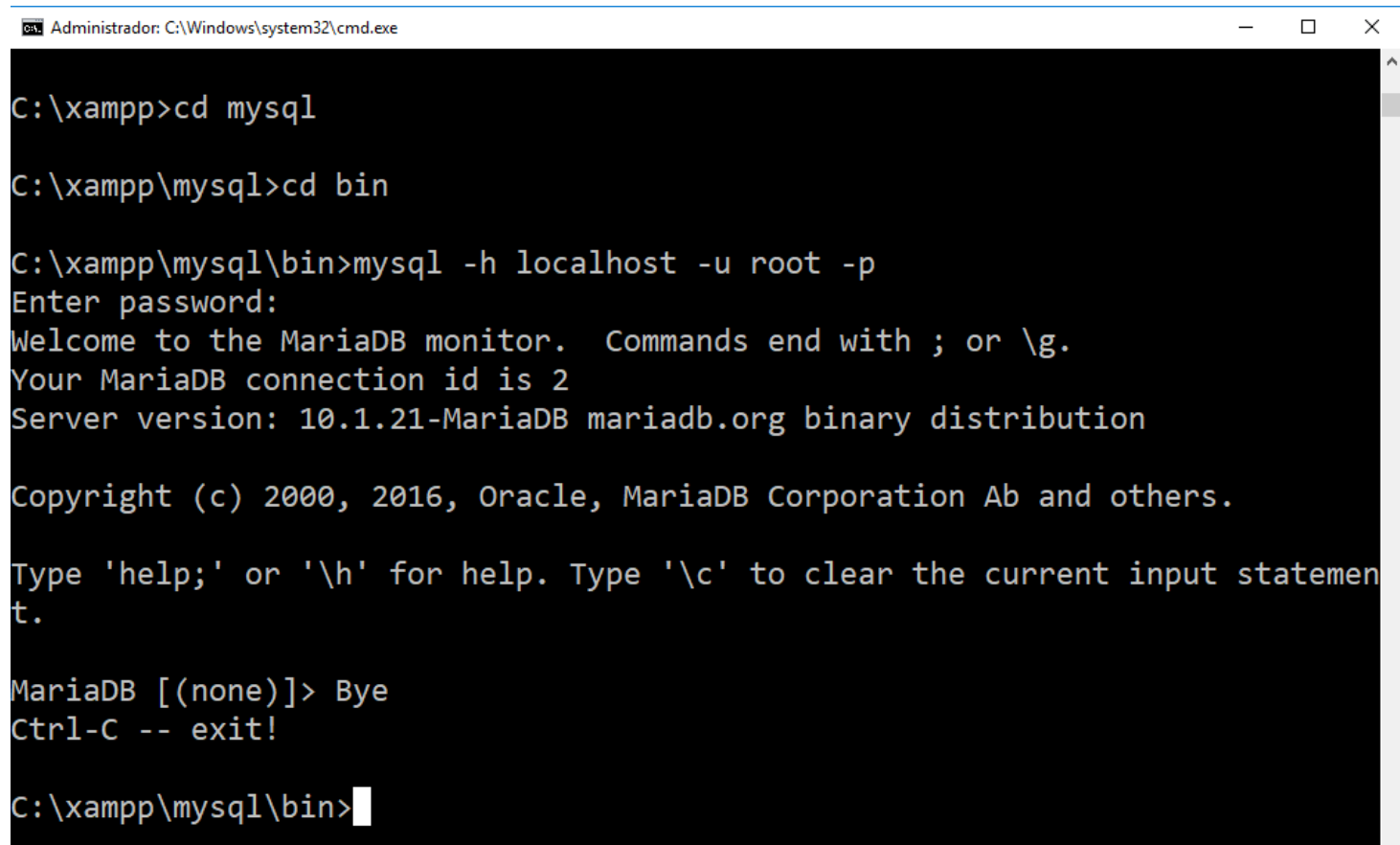
Saindo do MySQL

O comando para acessarmos o MySQL é:

Ctrl + C

ou

Exit

A screenshot of a Windows command prompt window titled "Administrador: C:\Windows\system32\cmd.exe". The window has a black background with white text. The command history shows the user navigating to the MySQL bin directory and attempting to exit the MySQL command-line interface. The MySQL prompt shows the user is logged in as root on localhost. The user enters Ctrl-C, which results in the MySQL prompt displaying "Bye" and "Ctrl-C -- exit!". The command prompt then returns to the user's directory.

```
Administrator: C:\Windows\system32\cmd.exe

C:\xampp>cd mysql

C:\xampp\mysql>cd bin

C:\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> Bye
Ctrl-C -- exit!

C:\xampp\mysql\bin>
```

Como criar um Novo Usuário

Na parte 1 do MySQL, nós fizemos toda a edição no MySQL como usuário root, com acesso total a todos as bases de dados. No entanto, em casos onde mais restrições possam ser necessárias, existem maneiras de criar usuários com permissões personalizadas.

Vamos começar criando um novo usuário dentro do shell do MySQL:

```
CREATE USER 'nouvousuario'@'localhost' IDENTIFIED BY 'password';
```

Exemplo:

```
create user 'senac'@'localhost' identified by 'senac';
```

Como criar um Novo Usuário

Infelizmente, nesse momento o novo usuário não tem permissão para fazer nada com as bases de dados. Na verdade, se o novo usuário tentar fazer login (com a senha password), ele não será capaz de chegar ao shell do MySQL.

Portanto, a primeira coisa a fazer é fornecer ao usuário o acesso às informações que ele vai precisar.

```
GRANT ALL PRIVILEGES ON * . * TO 'novousuario'@'localhost';
```

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 todas as bases de dados e tabelas.

Exemplo:

```
grant all privileges on * . * to 'senac'@'localhost';
```

Como criar um Novo Usuário

Uma vez finalizadas as permissões que você quer definir para os seus novos usuários, certifique-se sempre de recarregar todos os privilégios. Suas alterações agora estarão em vigor.

FLUSH PRIVILEGES;

Como conceder permissões

Aqui está uma pequena lista de outras possíveis permissões comuns que os usuários podem utilizar.

ALL PRIVILEGES- como vimos anteriormente, isso daria a um usuário do MySQL todo o acesso a uma determinada base de dados (ou se nenhuma base de dados for selecionada, todo o sistema);

CREATE- permite criar novas tabelas ou bases de dados;

DROP- permite deletar tabelas ou bases de dados;

DELETE- permite deletar linhas das tabelas;

INSERT- permite inserir linhas nas tabelas;

SELECT- permite utilizar o comando Select para ler bases de dados;

UPDATE- permite atualizar linhas das tabelas;

USAGE – Não altera privilégios; usado para tarefas administrativas na conta do usuário; **sem nenhum privilégio.*

GRANT OPTION- permite conceder ou revogar privilégios de outros usuários;

Como conceder permissões

Para dar uma permissão a um usuário específico, você pode utilizar esta estrutura:

```
GRANT [tipo de permissão] ON [nome da base de dados].[nome da tabela] TO '[nome do usuário]@'localhost';
```

Exemplo:

```
GRANT select, insert, update ON dbAcesso.* TO ti89@'localhost' WITH GRANT OPTION;
```

Se você quer dar a ele acesso a qualquer base de dados ou a qualquer tabela, certifique-se de colocar um asterisco (*) no lugar do nome da base de dados ou do nome da tabela.

Cada vez que você atualizar ou mudar uma permissão certifique-se de utilizar o comando.

Flush Privileges

Listar as permissões de acesso

Se você precisar visualizar uma permissão:

MariaDB [(none)]> **show grants for ti89@localhost;**

```
MariaDB [(none)]> select user,host from mysql.user;
+-----+-----+
| user | host |
+-----+-----+
| root | 127.0.0.1 |
| root | ::1 |
|      | localhost |
| pma  | localhost |
| root | localhost |
| ti89 | localhost |
+-----+-----+
6 rows in set (0.00 sec)

MariaDB [(none)]> show grants for ti89@localhost;
+-----+-----+
| Grants for ti89@localhost |
+-----+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'ti89'@'localhost' IDENTIFIED BY PASSWORD '*7F8C4740106D4AF3EF2AE7CB6FE17ACA9EAAC901' WITH GRANT OPTION |
+-----+-----+
1 row in set (0.00 sec)

MariaDB [(none)]>
```


Como conceder permissões

Se você precisar revogar uma permissão, a estrutura é quase idêntica a concedê-la:

REVOKE [tipo de permissão] ON [nome da base de dados].[nome da tabela] FROM '[nome do usuário]@'localhost';

Exemplo:

REVOKE DROP, DELETE ON dbAcesso.* FROM 'ti89'@'localhost';

Assim como você pode deletar bases de dados com o DROP, você pode utilizar o DROP para excluir um usuário completamente:

DROP USER 'demo'@'localhost';

Para testar seu novo usuário, faça logout digitando

quit

e faça login de volta com este comando no terminal:

mysql -u [nome do usuário]-p

Listar usuários

Se você precisar visualizar uma permissão:

MariaDB [(none)]> **select host,user from mysql.user;**

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select host,user from mysql.user;
+-----+-----+
| host      | user  |
+-----+-----+
| 127.0.0.1 | root  |
| ::1       | root  |
| localhost | pma    |
| localhost | root  |
| localhost | ti89   |
+-----+-----+
6 rows in set (0.00 sec)

MariaDB [(none)]> _
```

Visualizar as permissões

Se você precisar visualizar uma permissão:

```
show grants for 'root'@'localhost';
```

```
MariaDB [(none)]> show grants for 'root'@'localhost';
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH |
| GRANT PROXY ON ''@'%' TO 'root'@'localhost' WITH GRANT |
+-----+
2 rows in set (0.00 sec)
```

Criando usuários com grant

Pode-se criar usuários com o comando grant e dar as permissões adequadas.

MariaDB [(none)]> **grant usage on *.* to amarildo@localhost identified by '1234';**

```
MariaDB [(none)]> grant usage
-> on *.*
-> to
-> amarildo@localhost identified by '1234';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select user,host from mysql.user;
+-----+-----+
| user  | host  |
+-----+-----+
| root  | 127.0.0.1 |
| root  | ::1    |
|       | localhost |
| amarildo | localhost |
| pma    | localhost |
| root   | localhost |
| ti89   | localhost |
+-----+-----+
7 rows in set (0.00 sec)

MariaDB [(none)]> _
```

Criando usuários com grant

Visualizando os privilégios do usuário.

MariaDB [(none)]> **show grants for amarildo@localhost;**

```
MariaDB [(none)]> show grants for amarildo@localhost;
+-----+
| Grants for amarildo@localhost |
+-----+
| GRANT USAGE ON *.* TO 'amarildo'@'localhost' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7F932C8886EBFCF' |
+-----+
1 row in set (0.00 sec)

MariaDB [(none)]> _
```

Criando usuários com grant acesso total

Acesso total e fora de localhost.

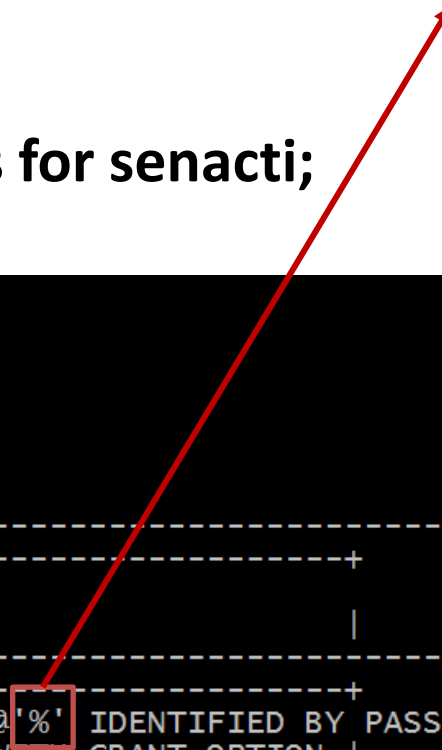
MariaDB [(none)]> **grant all on *.* to senacti identified by '1234' with grant option;**

MariaDB [(none)]> **show grants for senacti;**

```
MariaDB [(none)]> grant all
-> on *.*
-> to senacti identified by '1234'
-> with grant option;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show grants for senacti;
+-----+-----+-----+-----+
| Grants for senacti@% |
+-----+-----+-----+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'senacti'@'%' IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7F932C8886EBFCF' WITH GRANT OPTION |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [(none)]>
```



with grant option – o usuário terá acesso a criar outros usuários.

Criando banco de dados e inserindo permissões

MariaDB [(none)]> **create user senac@localhost;**

MariaDB [(none)]> **grant select,insert,update,delete on dbacesso.* to senac@localhost;**

MariaDB [(none)]> **show grants for senac@localhost;**

```
MariaDB [(none)]> create user senac@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant select,insert,update,delete
    -> on dbacesso.*
    -> to senac@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show grants for senac@localhost;
+-----+
| Grants for senac@localhost |
+-----+
| GRANT USAGE ON *.* TO 'senac'@'localhost' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `dbacesso`.* TO 'senac'@'localhost' |
+-----+
2 rows in set (0.00 sec)

MariaDB [(none)]> _
```

Revogando acesso ao banco de dados

MariaDB [(none)]> **show grants for senac@localhost;**

MariaDB [(none)]> **revoke delete on dbacesso.* from senac@localhost;**

MariaDB [(none)]> **show grants for senac@localhost;**

```
MariaDB [(none)]> show grants for senac@localhost;
+-----+
| Grants for senac@localhost |
+-----+
| GRANT USAGE ON *.* TO 'senac'@'localhost' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `dbacesso`.* TO 'senac'@'localhost' |
+-----+
2 rows in set (0.00 sec)

MariaDB [(none)]> revoke delete
    -> on dbacesso.*
    -> from senac@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show grants for senac@localhost;
+-----+
| Grants for senac@localhost |
+-----+
| GRANT USAGE ON *.* TO 'senac'@'localhost' |
| GRANT SELECT, INSERT, UPDATE ON `dbacesso`.* TO 'senac'@'localhost' |
+-----+
2 rows in set (0.00 sec)

MariaDB [(none)]> _
```


Revogando todos os acessos de um ou mais usuários

MariaDB [(none)]> **select** user,host from mysql.user;

MariaDB [(none)]> **revoke all, grant option** from senac@localhost, ti89@localhost;

MariaDB [(none)]> **show grants** for ti89@localhost;

```
MariaDB [(none)]> revoke all, grant option
-> from senac@localhost, ti89@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show grants for senac@localhost;
+-----+
| Grants for senac@localhost |
+-----+
| GRANT USAGE ON *.* TO 'senac'@'localhost' |
+-----+
1 row in set (0.00 sec)

MariaDB [(none)]> show grants for ti89@localhost;
+-----+
| Grants for ti89@localhost |
+-----+
| GRANT USAGE ON *.* TO 'ti89'@'localhost' IDENTIFIED BY PASSWORD '*7F8C4740106D4AF3EF2AE7CB6FE17ACA9EAAC901' |
+-----+
1 row in set (0.00 sec)

MariaDB [(none)]>
```