How to Display the Storage Engine in MariaDB?

written by sysadmin | 17 September 2025

If you are a user of the MariaDB database, you should know about the term storage engine.

Problem

How to display the storage engine in MariaDB?

Solution

A database engine (or storage engine) is the underlying software component that a database management system (DBMS) uses to create, read, update, and delete (CRUD) data from a database. To see a list of all available storage engines on the server, use the command below in the MariaDB prompt:

```
SHOW ENGINES;
```

and it will appear as shown in the image below:



Display all the storage engines in MariaDB

From the image above, you can see that InnoDB is the default storage engine in MariaDB. You can also use the command below to see the default storage engine in MariaDB:

```
SHOW GLOBAL VARIABLES LIKE 'default_storage_engine';
```

```
MariaDB [nodes]> SHOW GLOBAL VARIABLES LIKE 'default_storage_engine';
+------------------------+---------+
| Variable_name          | Value   |
+------------------------+---------+
| default_storage_engine | InnoDB |
+------------------------+---------+
1 row in set (0.006 sec)
```



Display the default storage engine in MariaDB

You can change the default storage engine by using the command below, which changes the default storage engine to MyISAM:

```
SET GLOBAL default_storage_engine='MyISAM';
```

Or you can use the below command if the session default storage engine supersedes the global default during this session:

```
SET SESSION default_storage_engine='MyISAM';
```

You should know that the storage engine is used per table and not per database, and to see it, you can use the command below:

```
SELECT TABLE_NAME, ENGINE
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'nodes';
```

and the result will look like below:

```
MariaDB [nodes]> SELECT TABLE_NAME, ENGINE
    -> FROM information_schema.TABLES
    -> WHERE TABLE_SCHEMA = 'nodes';
+------------------------+--------+
| TABLE_NAME             | ENGINE |
+------------------------+--------+
| flaggedrevs            | InnoDB |
| flaggedrevs_statistics | InnoDB |
| category               | InnoDB |
| babel                  | InnoDB |
+------------------------+--------+
4 rows in set (0.000 sec)
```



Display the storage engine used in the table

From the image above, it can be seen that all tables in the nodes database use InnoDB, so it can be said that the nodes database uses InnoDB. However, one database (schema) can have a table with a different engine, and the query below is the sample to create 2 tables that use different storage engines:

```
CREATE TABLE product (
    id INT PRIMARY KEY,
    name VARCHAR(100)
) ENGINE=InnoDB;

CREATE TABLE access_log (
    id INT PRIMARY KEY AUTO_INCREMENT,
    time DATETIME,
    user VARCHAR(50)
) ENGINE=MyISAM;
```

where the product table uses InnoDB and the table access_log uses MyISAM. If a database uses various storage engines and you want to know how many storage engines are used, you can use the query below:

```
SELECT ENGINE, COUNT(*) AS Number_of_tables
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'db_name'
GROUP BY ENGINE;
```

You can change the storage engine of a table by using the query below, for example, change it to MyISAM:

```
ALTER TABLE database_name.table_name ENGINE=MyISAM;
```

```
MariaDB [nodes]> SELECT TABLE_NAME, ENGINE
    -> FROM information_schema.TABLES
    -> WHERE TABLE_SCHEMA = 'nodes';
+-----------------------+--------+
| TABLE_NAME            | ENGINE |
+-----------------------+--------+
| flaggedrevs           | InnoDB |
| flaggedrevs_statistics | InnoDB |
| category              | InnoDB |
| babel                 | InnoDB |
+-----------------------+--------+
4 rows in set (0.000 sec)

MariaDB [nodes]> ALTER TABLE nodes.babel ENGINE=MyISAM;
Query OK, 19570 rows affected (0.076 sec)
Records: 19570  Duplicates: 0  Warnings: 0

MariaDB [nodes]> SELECT TABLE_NAME, ENGINE FROM information_schema.TABLES WHERE TABLE_SCHEMA = 'nodes';
+-----------------------+--------+
| TABLE_NAME            | ENGINE |
+-----------------------+--------+
| flaggedrevs           | InnoDB |
| flaggedrevs_statistics | InnoDB |
| category              | InnoDB |
| babel                 | MyISAM |
+-----------------------+--------+
4 rows in set (0.000 sec)
```

Change the storage engine in the table

Note

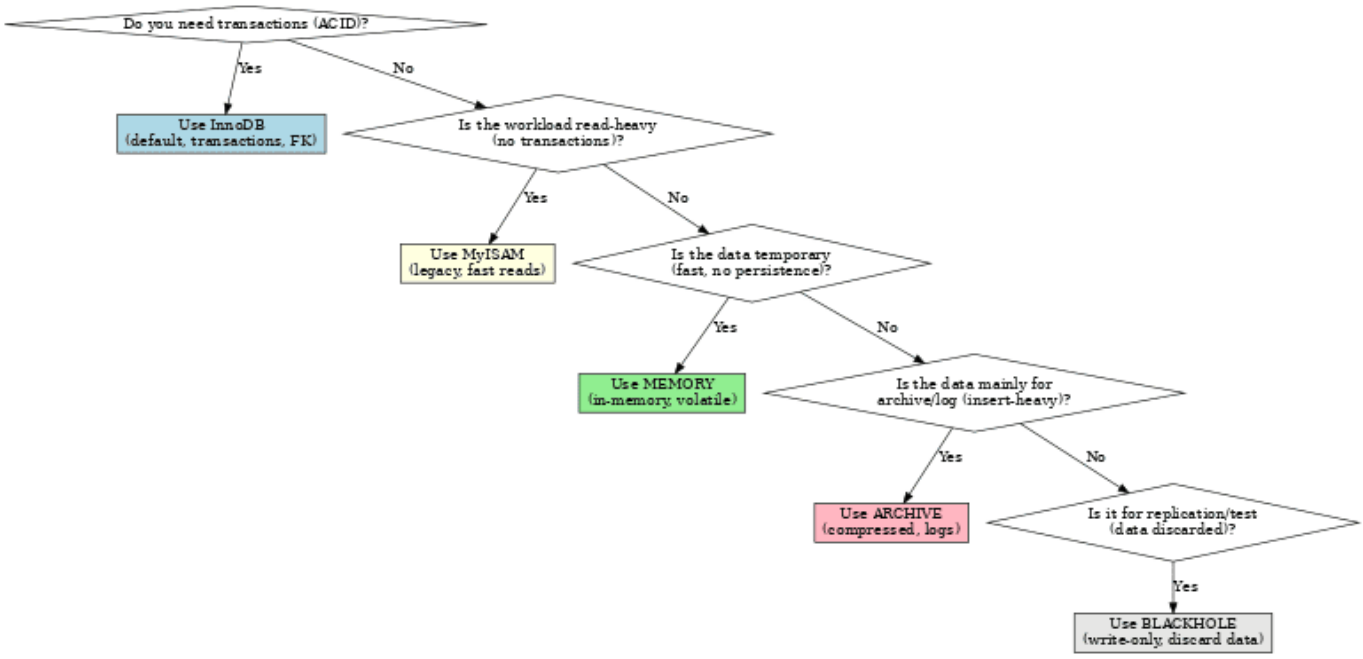
As explained above, there are many storage engines that you can use. But before you decide to use a storage engine, you should first find out the advantages and disadvantages of each storage engine, and to briefly see the differences between storage engines in the image below:

COMPARISON OF STORAGE ENGINES IN MARIADB/MYSQL

Feature / Engine	InnoDB (default)	MyISAM	MEMORY	ARCHIVE	BLACKHOLE
Transactions (ACID)	✅	❌	❌	❌	❌
Foreign Keys	✅	❌	❌	❌	❌
Locking	Row-level	Table-level	Table-level	Table-level	N/A
Crash recovery	Yes (redo /undo logs)	Limited (can lose data)	Data lost (in RAM)	Yes (insert only, compressed	Accepts writes but discards data
Read/Write speed	Balanced	Very fast on reads, slower writes	Extremely fast (in-memory)	Optimized for inserts only	None
Storage usage	General pur-pose apps, OLTP, most modern apps	Read-footprint workloads, legacy apps	RAM only	Very small (compressed)	Replication (pass-through), testing

Comparison between the storage engines

And below is a flowchart to determine the storage engines you will use:



Flowchart to choose the storage engine

However, as far as I know, among the many choices of storage engines, generally people use 2 types of storage engines, namely InnoDB and MyISAM. Maybe you can see the difference between the two through this site.

References

en.wikipedia.org
slideshare.net
medium.com
supportsages.com
mariadb.com
dev.mysql.com
managedserver.eu

How to Display All Databases Created by the User in MariaDB?

written by sysadmin | 17 September 2025

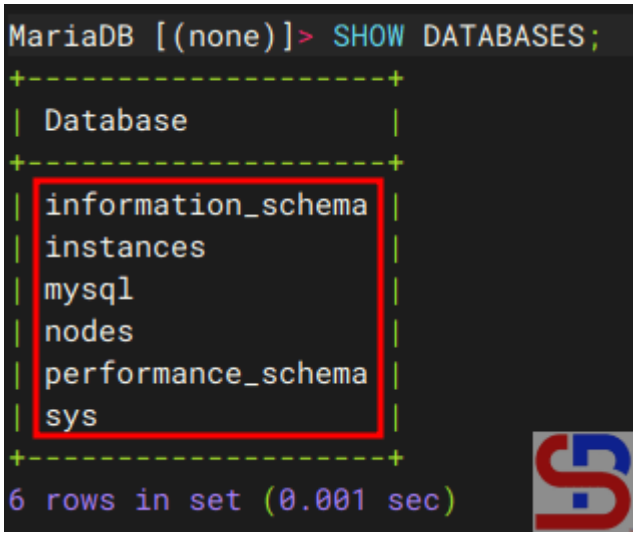
By default, in MariaDB, 4 databases are automatically created by the database system, namely **information_schema**, **mysql**, **performance_schema**, and **sys**. But I just want to display a database created by the user and not the default system databases.

Problem

How to display all databases created by the user in MariaDB?

Solution

If you use the **show databases** command after you access successfully accessing MariaDB, it will display all databases created by the user and default system databases, as shown below:

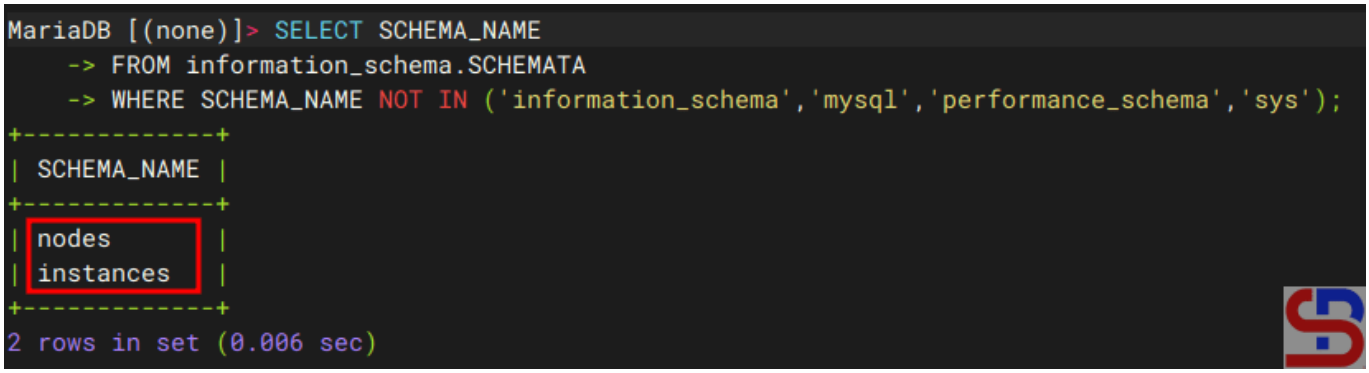


Display all databases

If you want to list only User-Created Databases, you can use a query as below:

```
SELECT SCHEMA_NAME
FROM information_schema.SCHEMATA
WHERE SCHEMA_NAME NOT IN
('information_schema','mysql','performance_schema','sys');
```

and it will display like the image below:



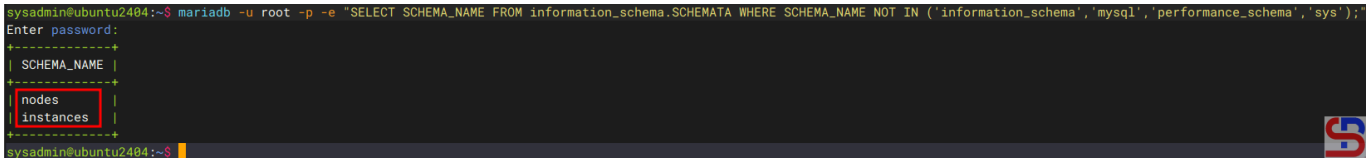
Display only User-Created Databases

Note

If you want to run the Linux command to display all databases created by the user only, use the command below:

```
mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA
WHERE SCHEMA_NAME NOT IN
('information_schema','mysql','performance_schema','sys');"
```

And you will see the result like the image below:



Display only User-Created Databases using the Linux command

References

stackoverflow.com
kinsta.com

How to Display the Total Size of the Entire MariaDB Database in the CLI?

written by sysadmin | 17 September 2025

I want to know the total size of the entire MariaDB database.

Problem

How to display the total size of the entire MariaDB database in the CLI?

Solution

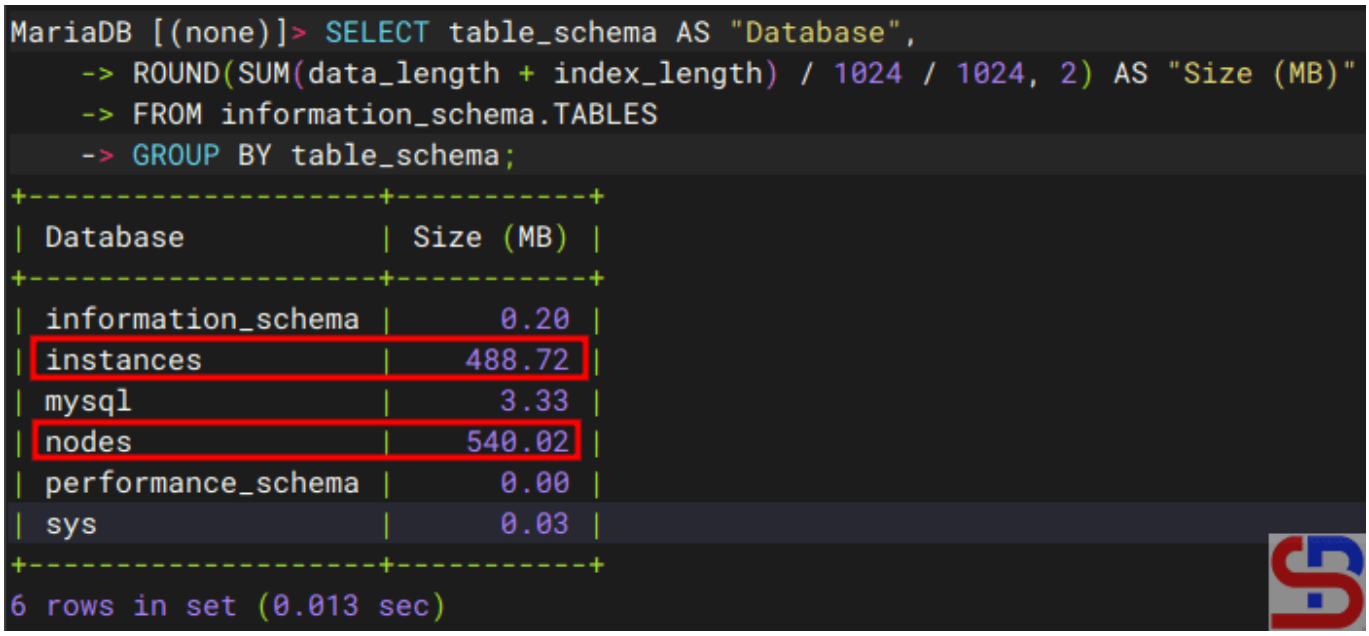
There are two methods for displaying the total size of the entire MariaDB database in the CLI:

1. Using a query

If you have entered the MariaDB database, you can use the command below to display the total size of a MariaDB database:

```
SELECT table_schema AS "Database",
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"
FROM information_schema.TABLES
GROUP BY table_schema;
```

The above query will display the total size of the entire MariaDB database in MegaBytes (MB) as shown in the image below:



Display the total size of the databases using the query in Megabytes

If you want to display in Gigabytes (GB), use the command below:

```
SELECT table_schema AS "Database",
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS "Size (GB)"
FROM information_schema.TABLES
GROUP BY table_schema;
```

so that it will display as shown in the image below:

```
MariaDB [(none)]> SELECT table_schema AS "Database",
    -> ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS "Size (GB)"
    -> FROM information_schema.TABLES
    -> GROUP BY table_schema;
+--------------------+-----------+
| Database           | Size (GB) |
+--------------------+-----------+
| information_schema |      0.00 |
| instances          |      0.48 |
| mysql              |      0.00 |
| nodes              |      0.53 |
| performance_schema |      0.00 |
| sys                |      0.00 |
+--------------------+-----------+
6 rows in set (0.012 sec)
```

Display the total size of the databases using the query in GigaBytes

2. Using Linux Command

If you want to display the total size of the entire MariaDB database using Linux commands, use the command below:

```
mariadb -u root -p -e "
SELECT table_schema AS 'Database',
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
```

If you run the above command, it will display like the image below:

```
root@ubuntu2404:~# mariadb -u root -p -e "
SELECT table_schema AS 'Database',
       ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
Enter password:
+--------------------+-----------+
| Database           | Size (MB) |
+--------------------+-----------+
| information_schema |      0.20 |
| instances          |    488.72 |
| mysql              |      3.33 |
| nodes              |    540.02 |
| performance_schema |      0.00 |
| sys                |      0.03 |
+--------------------+-----------+
root@ubuntu2404:~#
```

Display the total size of the databases using the Linux command in Megabytes

If you want to display it in GigaBytes, use the command below:

```
mariadb -u root -p -e "
SELECT table_schema AS 'Database',
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Size (GB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
```

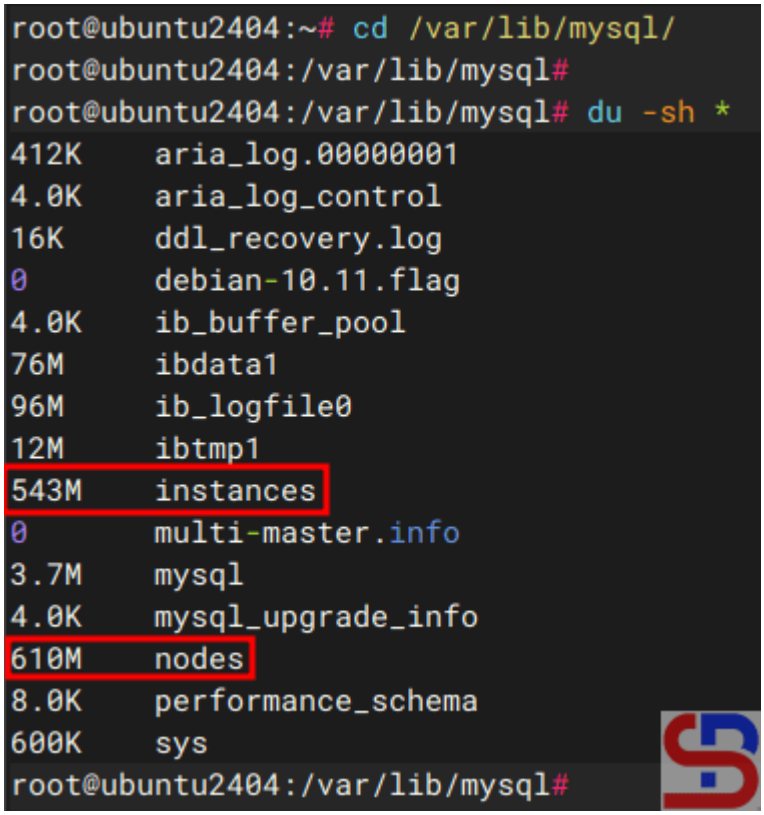
and it will display like the image below:

```
root@ubuntu2404:~# mariadb -u root -p -e "
SELECT table_schema AS 'Database',
       ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Size (GB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
Enter password:
+--------------------+-----------+
| Database           | Size (GB) |
+--------------------+-----------+
| information_schema |      0.00 |
| instances          |      0.48 |
| mysql              |      0.00 |
| nodes              |      0.53 |
| performance_schema |      0.00 |
| sys                |      0.00 |
+--------------------+-----------+
root@ubuntu2404:~#
```

Display the total size of the databases using the Linux command in Gigabytes

Or in a short time, you can use the Linux command below to see the total size of the entire database in MariaDB:

```
cd /var/lib/mysql/
du -sh *
```



Display the total size of the databases using the Linux command

Note

If you want to display the total size of the entire MariaDB database in KiloBytes, then in the **ROUND(SUM(data_length + index_length)** section, simply divide it by 1024 so that it becomes as follows:

```
mariadb -u root -p -e "
SELECT table_schema AS "Database",
ROUND(SUM(data_length + index_length) / 1024 , 2) AS "Size (KB)"
FROM information_schema. TABLES
GROUP BY table_schema;"
```

And if you want to display in TeraBytes size, then in the **ROUND(SUM(data_length + index_length)** section, divide by 1024 4 times, so that the command is as follows:

```
mariadb -u root -p -e "
SELECT table_schema AS 'Database',
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024 / 1024 , 2) AS
'Size (KB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
```

References

a2hosting.com
database.guide
tecmint.com

How to Reset MariaDB Root Password?

written by sysadmin | 17 September 2025

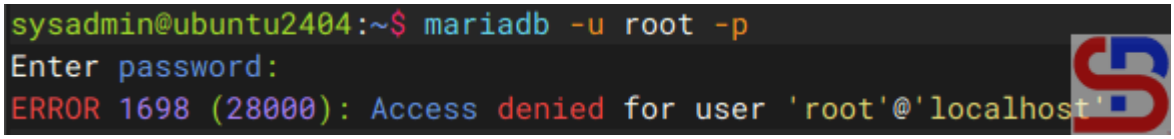
I want to access the MariaDB database using the root user, but I forgot the password of the root user.

Problem

How to reset MariaDB root password?

Solution

When I insert my password root to access MariaDB, here is the error:



Error when accessing MariaDB

Here are the steps to reset the root password in MariaDB:

1. Stop the service

Use the command below to stop MariaDB’s service:

```
sudo systemctl stop mariadb
```

2. Running Mariadb in Safe Mode

Run the command below to run the MariaDB service without access rights and networks:

```
sudo mysqld_safe --skip-grant-tables --skip-networking &
```

3. Change the password

Enter MariaDB using the command:

```
sudo mariadb -uroot -p
```

Press the **Enter** button and after that, type the commands below, and I use the **qwerty** password as my new password:

```
FLUSH PRIVILEGES;
ALTER USER root@localhost IDENTIFIED BY 'qwerty';
FLUSH PRIVILEGES;
exit
```

4. Stop MariaDB

Use the command below to stop MariaDB’s service:

```
sudo mysqladmin -u root -p shutdown
```

Enter the password you just created if you are asked to enter a password.

5. Running the Service

Run the command below to turn on MariaDB's service:

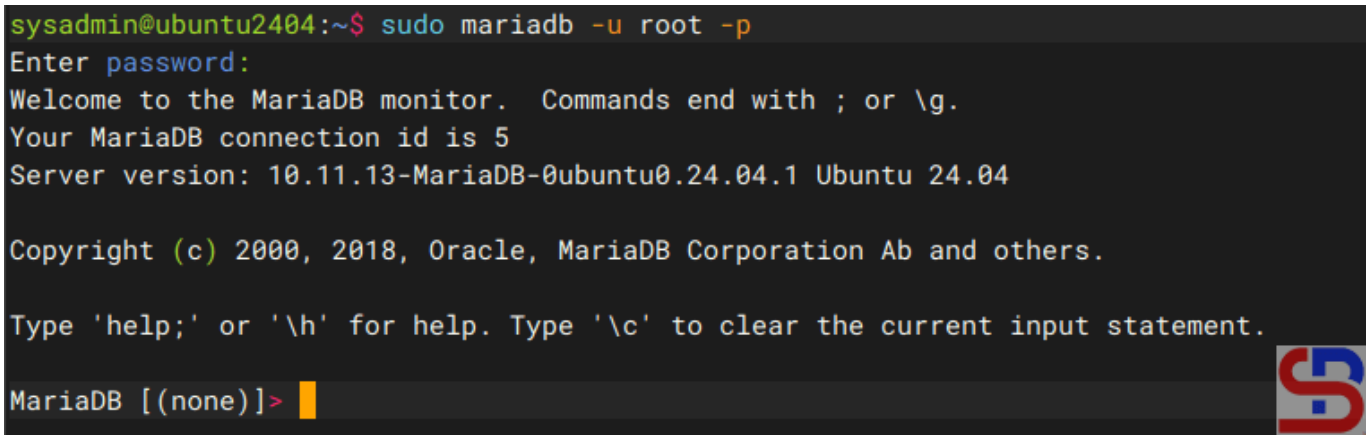
```
sudo systemctl start mariadb
```

6. Try a new password

Enter MariaDB by using the password that you created previously using the command below:

```
sudo mariadb -u root -p
```

Insert your new password, and if you type the correct password, then you should be able to enter Mariadb as in the image below:



Test your new password to access MariaDB

Note

Maybe as an alternative to storing passwords for the root user in MariaDB, you can store the password in a file, but using an encrypted Vim editor.

References

musaamin.web.id
serverspace.io
vexxhost.com

How to Manage the User(s) in MariaDB?

written by sysadmin | 17 September 2025

The previous article already explained how to create a database and a table in MariaDB. This article will explain how to manage the user(s) in MariaDB.

Problem

How to manage the user(s) in MariaDB?

Solution

Here are the commands to manage user(s) in MariaDB:

INFO

The use of capital letters in this article is only to distinguish between original commands from MariaDB and data from the user. You don't have to use the capital letters when running these commands, but you can use all lowercase letters.

1. Display all users

Type the command below to display all the users in MariaDB:

```
SELECT User,Host FROM mysql.user;
```

```
MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
3 rows in set (0.007 sec)
```



Display all users in MariaDB

You can see in the picture above that by default, there are only 3 users in MariaDB.

2. Create a user with a password

Use the format below to create a user with a password:

```
CREATE USER 'username'@'ip_address' IDENTIFIED BY 'password_user';
```

For example, If you want to create a user with the name james and its password 123456, then type the command below:

```
CREATE USER 'james'@'localhost' IDENTIFIED BY '123456';
```

If you want to create a John user with a qwerty password but the user can only access the database via the IP with subnet 192.168.56.0/24, type the command below:

```
CREATE USER 'john'@'192.168.56.%' IDENTIFIED BY 'qwerty';
```

But if you also want to create a judith user with a password 1q2w3e4r and the user can access the database from any network, then type the command below:

```
CREATE USER 'judith'@'%' IDENTIFIED BY '1q2w3e4r';
```

```
MariaDB [(none)]> CREATE USER 'james'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.051 sec)

MariaDB [(none)]> CREATE USER 'john'@'192.168.56.%' IDENTIFIED BY 'qwerty';
Query OK, 0 rows affected (0.043 sec)

MariaDB [(none)]> CREATE USER 'judith'@'%' IDENTIFIED BY '1q2w3e4r';
Query OK, 0 rows affected (0.047 sec)

MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| judith      | %            |
| john        | 192.168.56.% |
| james       | localhost    |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
6 rows in set (0.004 sec)
```

Create the users

To see the options for this command, you can go to this page.

4. Rename user

Use the format below to change the user name:

```
RENAME USER 'username1'@'ip_address' TO 'username2'@'ip_address'
```

For example, you want to change the name of james to bob by typing the command below:

```
RENAME USER 'james'@'localhost' TO 'bob'@'localhost';
```

```
MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| judith      | %            |
| john        | 192.168.56.% |
| james       | localhost    |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
6 rows in set (0.003 sec)

MariaDB [(none)]> RENAME USER 'james'@'localhost' TO 'bob'@'localhost';
Query OK, 0 rows affected (0.043 sec)

MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| judith      | %            |
| john        | 192.168.56.% |
| bob         | localhost    |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
6 rows in set (0.003 sec)
```

Rename the user

You can also change the IP address using the command:

```
RENAME USER 'bob'@'localhost' TO 'bob'@'192.168.56.%';
```

```
MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| judith      | %            |
| john        | 192.168.56.% |
| bob         | localhost    |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
6 rows in set (0.005 sec)

MariaDB [(none)]> RENAME USER 'bob'@'localhost' TO 'bob'@'192.168.56.%';
Query OK, 0 rows affected (0.046 sec)

MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| judith      | %            |
| bob         | 192.168.56.% |
| john        | 192.168.56.% |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
6 rows in set (0.003 sec)
```

Rename the IP address

To see the options for this command, you can go to this page.

5. Alter user

Use the format below to change the user in MariaDB:

```
ALTER USER 'username' option1 option2 ... optionN;
```

For example, if you want to change the user james' password, then use the command below:

```
ALTER USER 'james'@'localhost' IDENTIFIED BY 'qwerty';
```

```
MariaDB [(none)]> ALTER USER 'james'@'localhost' IDENTIFIED BY 'qwerty';
Query OK, 0 rows affected (0.006 sec)
```



Change the password using the alter command

To see the options for this command, you can go [to this page](#).

6. Grant a user

If you only make a user in Mariadb without giving access to the user, then the user will not be able to enter the existing database in Mariadb. By default, only the root user gets access to all databases in Mariadb. To add a user to have access, use the format below:

```
GRANT option1 ON option2 TO 'username'@'ip_address;
```

Option1 is for privileges options at a database or a table level, and option2 is for which database or table the user can access by the user. To see the options for this command, you can go [to this page](#). For example, the user bob can only access the db_office database and then use the command below:

```
GRANT ALL ON db_office.* TO 'bob'@'192.168.56.%';
```

If you want to provide access to the user john to only be able to do the **select** command for the employee table in the db_office database, then use the command below:

```
GRANT SELECT ON db_office.employees TO 'john'@'192.168.56.%';
```

and provide a judith user to access the entire database, use the command below:

```
GRANT ALL ON *.* TO 'judith'@'%';
```

You can also combine the grant command by giving a password to a user by typing the command below:

```
GRANT ALL ON db_office.* TO 'richard'@'192.168.56.%' IDENTIFIED BY 'qwerty';
```

To see the grant of a user, for example, a judith user, type the command below:

```
SHOW GRANTS for 'judith';
```

```
MariaDB [db_office]> SHOW GRANTS for 'judith';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for judith@%                                                                                             |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO `judith`@`%` IDENTIFIED BY PASSWORD '*DB469070DB0AD0CA0B93040D166D7FC4713D6961' |
+-----------------------------------------------------------------------------------------------------------------+
1 row in set (0.000 sec)
```

Display grant for a user

Warning

If you want to see the grant of a user but you do not enter the IP address of the user's host, as in the picture above, then by default, MariaDB will assume '%' as the host. Therefore, it is recommended that you type the IP address of the user to display the grant status of the user.

If you want to see what grant access is given to a user but forget each IP host address from the user-user in MariaDB, then use this command:

```
select distinct concat('SHOW GRANTS FOR ', QUOTE(user), '@', QUOTE(host), ';') as query from mysql.user;
```

```
MariaDB [db_office]> select distinct concat('SHOW GRANTS FOR ', QUOTE(user), '@', QUOTE(host), ';') as query from mysql.user;
+-----------------------------------------+
| query                                   |
+-----------------------------------------+
| SHOW GRANTS FOR 'judith'@'%';           |
| SHOW GRANTS FOR 'bob'@'192.168.56.%';   |
| SHOW GRANTS FOR 'john'@'192.168.56.%';  |
| SHOW GRANTS FOR 'richard'@'192.168.56.%'; |
| SHOW GRANTS FOR 'mariadb.sys'@'localhost'; |
| SHOW GRANTS FOR 'mysql'@'localhost';    |
| SHOW GRANTS FOR 'root'@'localhost';     |
+-----------------------------------------+
7 rows in set (0.009 sec)
```

Display all grants for each user

7. Make a role

By default, if you have many users and sometimes these users have the same access, it is recommended to create a role. A role bundles many privileges together. Use the format below to create a role:

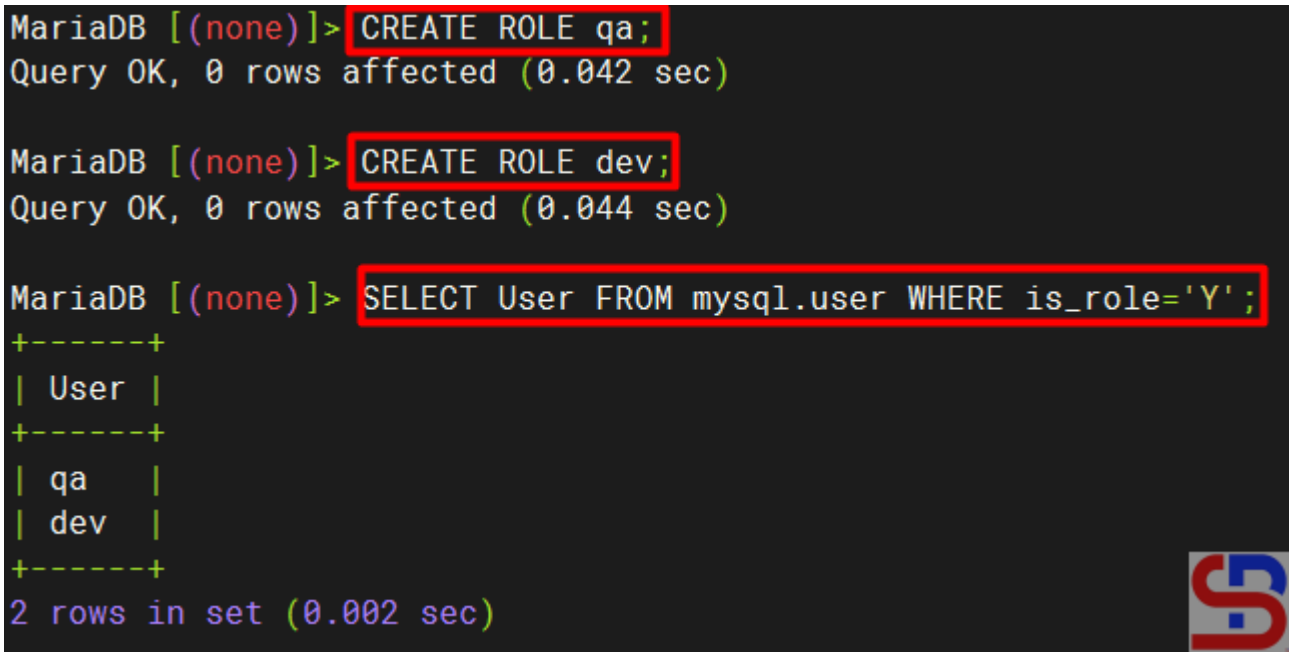
```
CREATE ROLE role_name;
```

To see the options in this command, please go to this page. For example, if you want to make a qa and a dev role in MariaDB, then use the command below to make the role:

```
CREATE ROLE qa;
CREATE ROLE dev;
```

To see all the roles in MariaDB, use the command below:

```
SELECT User FROM mysql.user WHERE is_role='Y';
```



Create the role

After that, use the grant command to access the db_office database based on the roles as the command below:

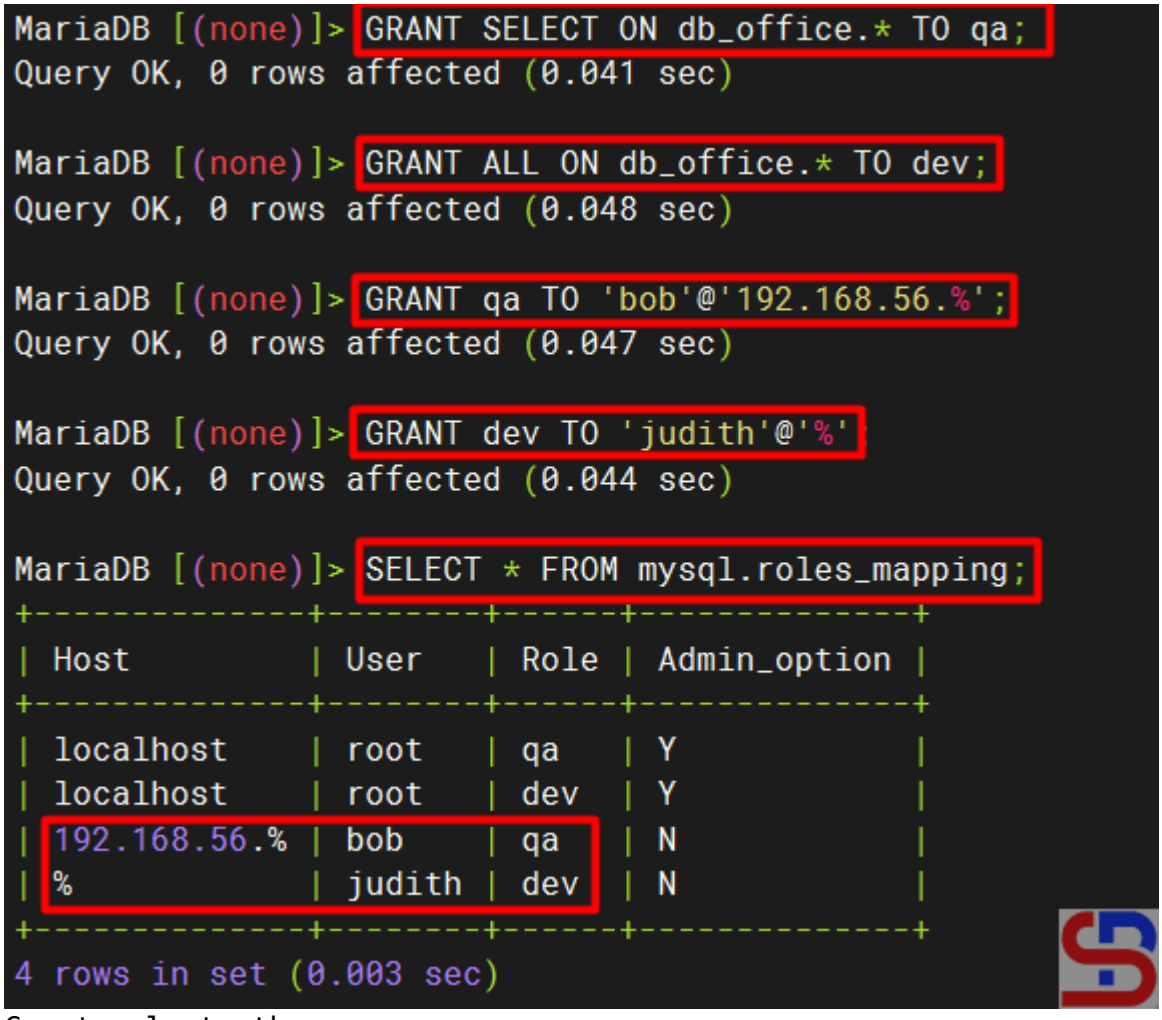
```
GRANT SELECT ON db_office.* TO qa;
GRANT ALL ON db_office.* TO dev;
```

Then enter the user bob into the qa role and judith into the dev role with the command below:

```
GRANT qa TO 'bob'@'192.168.56.%';
GRANT dev to 'judith'@'%';
```

To see users who have entered into the roles in Mariadb, use the command below:

```
SELECT * FROM mysql.roles_mapping;
```



Grant role to the user

8. Delete access

If you want to delete access to a user, whether it’s a role or a grant. Use the format below if you want to delete a user’s role:

```
REVOKE role_name FROM 'user'@'ip_address';
```

For example, if you want to delete the qa role from the user bob, use the command below:

```
REVOKE qa FROM 'bob'@'192.168.56.%';
```

```
MariaDB [db_office]> SELECT * FROM mysql.roles_mapping;
+--------------+--------+------+--------------+
| Host         | User   | Role | Admin_option |
+--------------+--------+------+--------------+
| localhost    | root   | qa   | Y            |
| localhost    | root   | dev  | Y            |
| 192.168.56.% | bob    | qa   | N            |
| %            | judith | dev  | N            |
+--------------+--------+------+--------------+
4 rows in set (0.002 sec)

MariaDB [db_office]> REVOKE qa FROM 'bob'@'192.168.56.%';
Query OK, 0 rows affected (0.045 sec)

MariaDB [db_office]> SELECT * FROM mysql.roles_mapping;
+-----------+--------+------+--------------+
| Host      | User   | Role | Admin_option |
+-----------+--------+------+--------------+
| localhost | root   | qa   | Y            |
| localhost | root   | dev  | Y            |
| %         | judith | dev  | N            |
+-----------+--------+------+--------------+
3 rows in set (0.002 sec)

MariaDB [db_office]>
```

Revoke a user's role

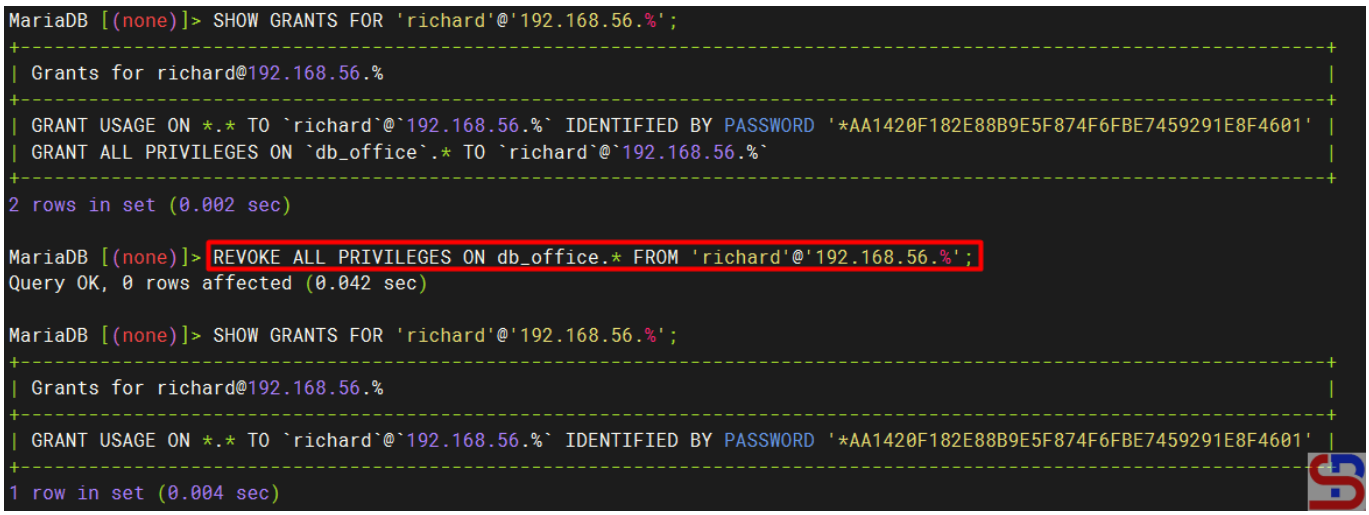
Use the format below if you want to delete a grant:

```
REVOKE option1 ON option2 FROM 'user'@'ip_address';
```

Option1 is for privileges options at a database or a table level, and option2 is for which database or table. If you want to see a deeper explanation of this command, go to this

page. For example, if you want to delete richard's grant in the db_office database, then use the command below:

```
REVOKE ALL PRIVILEGES ON db_office.* FROM 'richard'@'192.168.56.%';
```



Revoke the user's grant

9. Delete a user

To delete a user, use the format below:

```
DROP USER username;
```

If you want to see a deeper explanation of this command, please go to this page. For example, if you want to delete the judith user in MariaDB, then use the command below:

```
DROP USER judith;
```

```
MariaDB [(none)]> SELECT USER,HOST FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| dev         |              |
| qa          |              |
| judith      | %            |
| bob         | 192.168.56.% |
| john        | 192.168.56.% |
| richard     | 192.168.56.% |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
9 rows in set (0.004 sec)

MariaDB [(none)]> DROP USER judith;
Query OK, 0 rows affected (0.059 sec)

MariaDB [(none)]> SELECT USER,HOST FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| dev         |              |
| qa          |              |
| bob         | 192.168.56.% |
| john        | 192.168.56.% |
| richard     | 192.168.56.% |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
8 rows in set (0.004 sec)
```

Delete the user

But if you find a user who has 2 names that are the same, but the IP host address is different, then you must use the format below:

```
DROP USER 'username'@'ip_address';
```

For example, you want to delete one of the user bobs that

has an IP 192.168.56.%, Then use the command below:

```
DROP USER 'bob'@'192.168.56.%';
```

```
MariaDB [(none)]> SELECT USER,HOST FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| dev         |              |
| qa          |              |
| bob         | %            |
| bob         | 192.168.56.% |
| john        | 192.168.56.% |
| richard     | 192.168.56.% |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
9 rows in set (0.003 sec)

MariaDB [(none)]> DROP USER 'bob'@'192.168.56.%';
Query OK, 0 rows affected (0.046 sec)

MariaDB [(none)]> SELECT USER,HOST FROM mysql.user;
+-------------+--------------+
| User        | Host         |
+-------------+--------------+
| dev         |              |
| qa          |              |
| bob         | %            |
| john        | 192.168.56.% |
| richard     | 192.168.56.% |
| mariadb.sys | localhost    |
| mysql       | localhost    |
| root        | localhost    |
+-------------+--------------+
8 rows in set (0.004 sec)
```

Delete the user with a certain IP

Note

To provide grant access to one of the users, it must first be asked what his needs are for accessing a database so that the MariaDB database can be more secure.

References

mariadb.com
gist.github.com
cyberciti.biz
severalnines.com

How to Manage a Database and its Table(s) in MariaDB?

written by sysadmin | 17 September 2025

After installing MariaDB on your Linux server, you must know some basic MariaDB commands.

Problem

How to manage a database and its table(s) in MariaDB?

Solution

Below are the basic commands of MariaDB to manage a database and its table(s):

1. Access to the MariaDB database

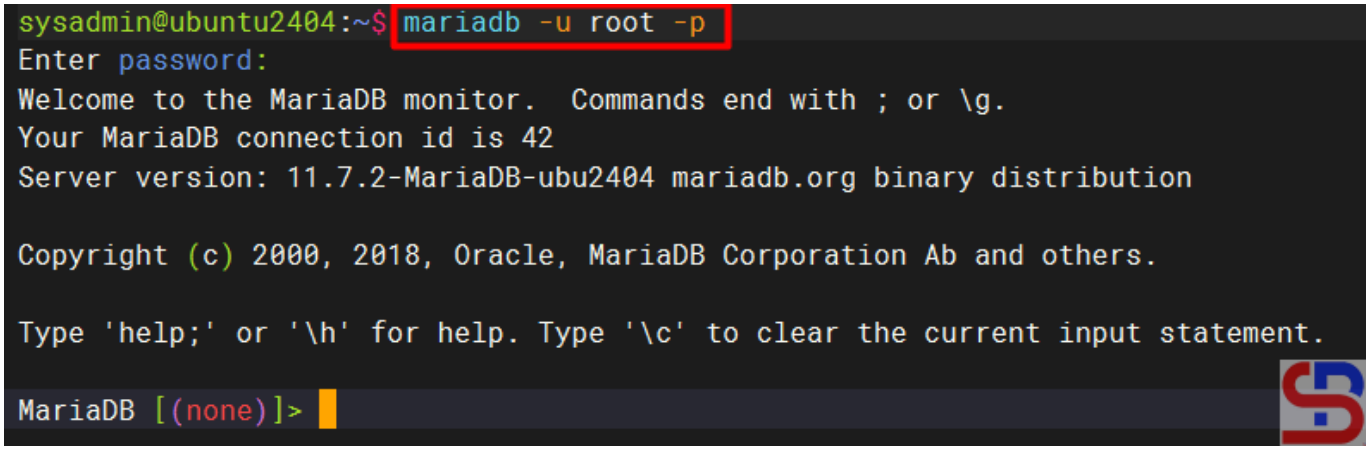
Use the format below to access MariaDB:

```
mariadb -h ip_address -u username -p
```

If you access MariaDB from the server using the root user directly, type the command below:

```
mariadb -u root -p
```

Enter the password for the root in MariaDB, and if the password is appropriate then you can access MariaDB as in the image below:



Access to MariaDB

INFO

The use of capital letters in this article is only to distinguish between original commands from MariaDB and data from the user. You don't have to use the capital letters when running these commands, but you can use all lowercase letters.

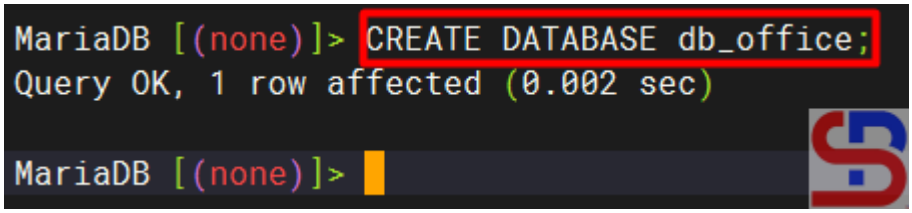
2. Creating a new database

Use the format below to create a new database:

```
CREATE DATABASE database_name;
```

You can see the options for this command on this page. For example, if you want to create a new database called **db_office**, use the command below;

```
CREATE DATABASE db_office;
```

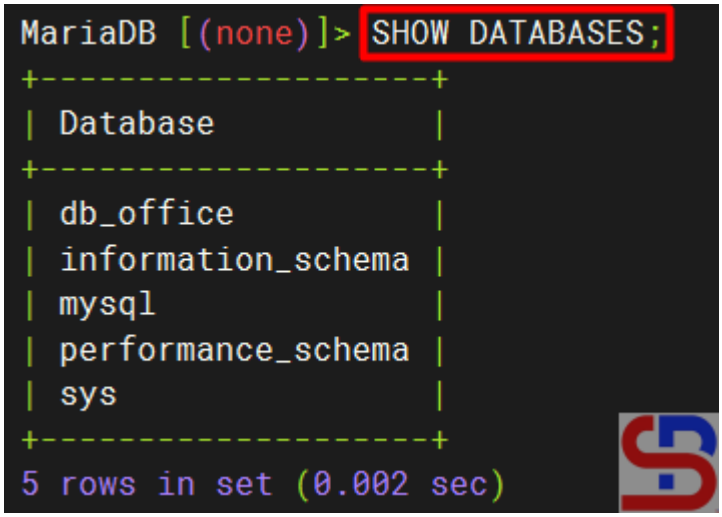


Create a new database

3. See the entire database

Type the command below to see all the databases stored in MariaDB:

```
SHOW DATABASES;
```



Display all databases

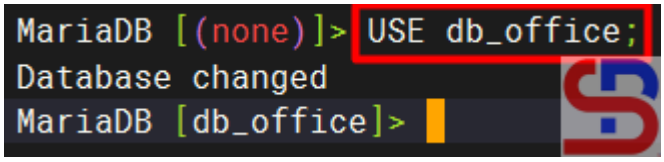
4. Select a database

Use the format below to select the database you want to access:

```
USE db_name;
```

For example, if you want to access the db_office database, then type the command below:

```
USE db_office;
```



Select the database

5. Create a table

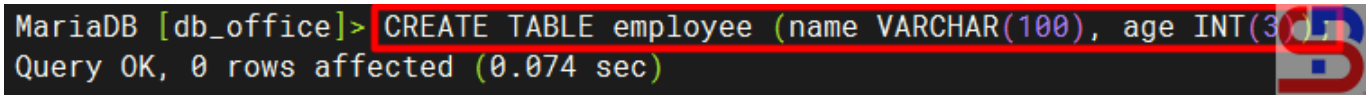
Use the format below to create a new table:

```
CREATE TABLE table_name (name_of_column1 column_data_type1, name_of_column2
```

```
column_data_type2, ...);
```

You can see the options for this command on this page. Type the command below to create an employee table:

```
CREATE TABLE employee (name varchar (100), age int (3));
```



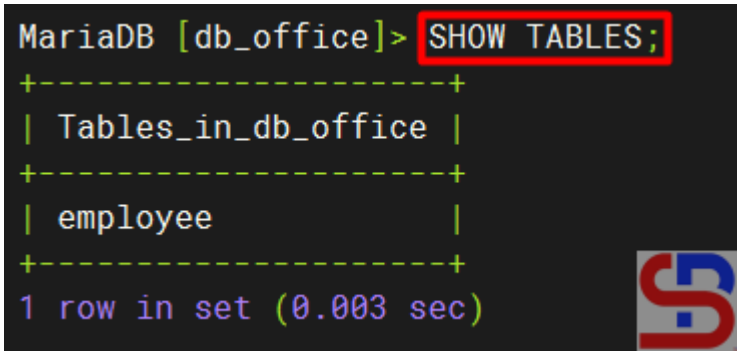
Create a new table

You can see data types that can be used on this page.

6. Display the entire table(s)

Use the command below to display the entire table in a database:

```
SHOW TABLES;
```



Show all tables

7. Display the table structure

Use the command below to display the table structure:

```
DESC table_name;
```

For example, if you want to see the employee table structure, then use the command below:

```
DESC employee;
```

```
MariaDB [db_office]> DESC employee;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name  | varchar(100) | YES  |     | NULL    |       |
| age   | int(3)       | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
2 rows in set (0.004 sec)
```

Display a table structure

8. Add a column

Use the format below to make a column in table:

```
ALTER TABLE db_name ADD COLUMN column_name type (nnn);
```

You can see the options for this command on this page. For example, if you want to add to the city column in the employee table, use the command below:

```
ALTER TABLE employee ADD COLUMN city VARCHAR (100);
```

```
MariaDB [db_office]> ALTER TABLE employee ADD COLUMN city VARCHAR (100);
Query OK, 0 rows affected (0.063 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [db_office]> DESC employee;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name  | varchar(100) | YES  |     | NULL    |       |
| age   | int(3)       | YES  |     | NULL    |       |
| city  | varchar(100) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

MariaDB [db_office]>
```

Add a new column

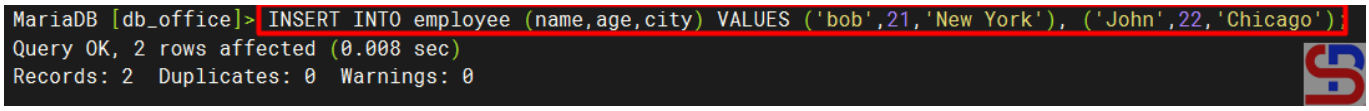
9. Insert data into the table

Use the format below to enter data in a table:

```
INSERT INTO table_name (Column1, Column2,…, ColumnN) VALUES (Value1,…ValueN), (Value1,…ValueN);
```

You can see the options for this command on this page. Type the command below if you want to insert 2 data to the employee table:

```
INSERT INTO employee (name,age,city) VALUES ('bob',21,'New York'), ('John',22,'Chicago');
```



Insert data into the table

INFO

If you want to insert a value in the form of a number, the number does not have to be flanked with an apostrof (‘…’) sign, whereas if it is a character or a combination of characters and numbers, it must be flanked with an apostrof (‘…’).

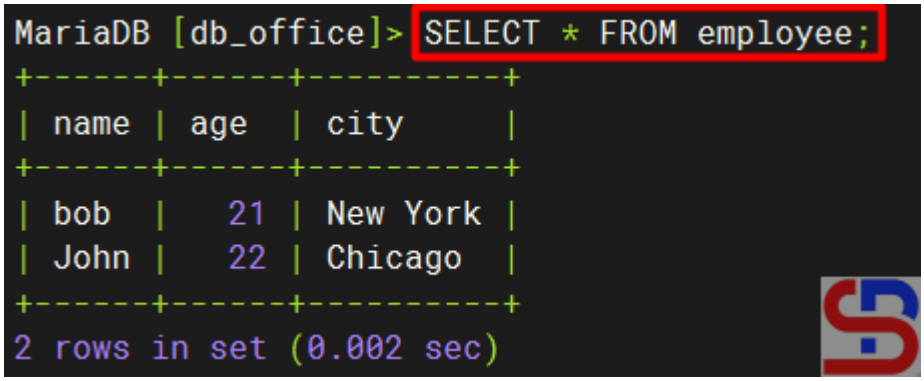
10. Displays data in a table

Use the format below to display all data in a table:

```
SELECT option1 FROM table_name option2;
```

You can see the options for this command on this page. For example, use the command below to display all data in the employee table:

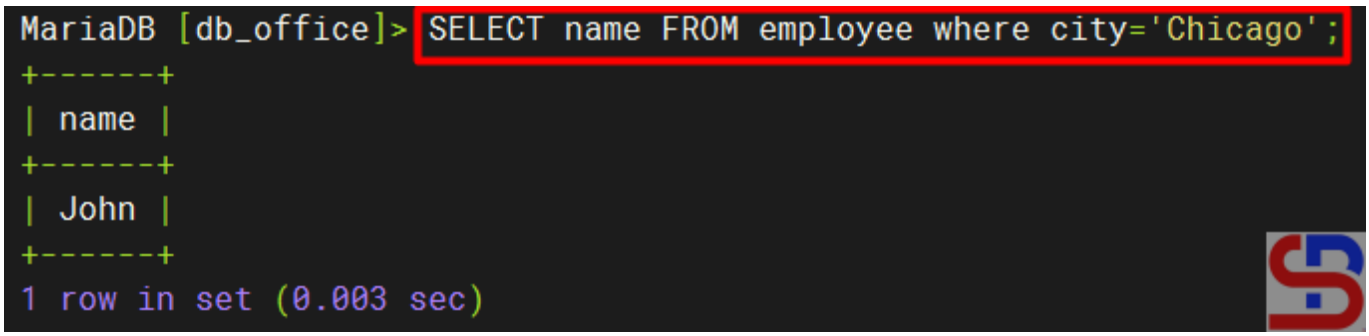
```
SELECT * FROM employee;
```



Display all the data

Or, if you want to display the name of the user who lives in the city of Chicago then use the command below:

```
SELECT name FROM employee where city='Chicago';
```



Display the data with a condition

11. Update data

Use the format below to update data in a table:

```
UPDATE table_name SET columnX=valueX WHERE columnY=valueY;
```

You can see the options for this command on this page. For example, if you want to update the age of the employee named Bob, use the command below:

```
UPDATE employee SET age=23 WHERE name='bob';
```

```
MariaDB [db_office]> SELECT * FROM employee;
+------+------+----------+
| name | age  | city     |
+------+------+----------+
| bob  |   21 | New York |
| John |   22 | Chicago  |
+------+------+----------+
2 rows in set (0.004 sec)

MariaDB [db_office]> UPDATE employee SET age=23 WHERE name='bob';
Query OK, 1 row affected (0.046 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [db_office]> SELECT * FROM employee;
+------+------+----------+
| name | age  | city     |
+------+------+----------+
| bob  |   23 | New York |
| John |   22 | Chicago  |
+------+------+----------+
2 rows in set (0.001 sec)
```

Update data

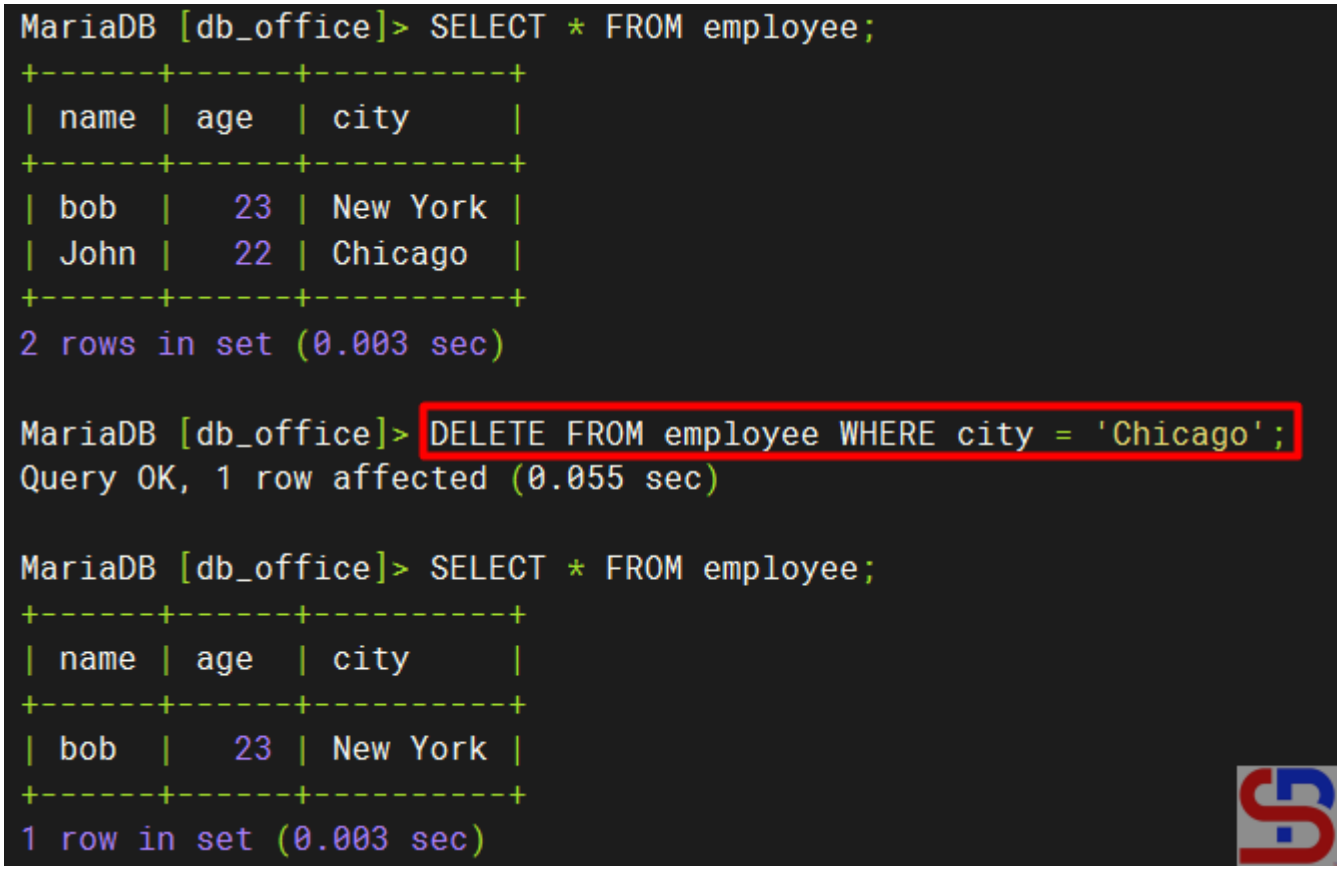
12. Delete Data

Use the format below to delete one or more rows of a table:

```
DELETE FROM table_name WHERE column=value;
```

You can see the options for this command on this page. For example, if you want to delete the data where the user is in Chicago, then use the command below:

```
DELETE FROM employee WHERE city='Chicago';
```



Delete the data

13. Delete the column

Use the format below to make changes in the table:

```
ALTER TABLE db_name DROP COLUMN column_name;
```

If you want to delete the column, for example, city, you can use the following command:

```
ALTER TABLE employee DROP COLUMN city;
```

```
MariaDB [db_office]> DESC employee;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(100) | YES  |     | NULL    |       |
| age   | int(3)      | YES  |     | NULL    |       |
| city  | varchar(100) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)

MariaDB [db_office]> ALTER TABLE employee DROP COLUMN city;
Query OK, 0 rows affected (0.022 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [db_office]> DESC employee;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(100) | YES  |     | NULL    |       |
| age   | int(3)      | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.001 sec)

MariaDB [db_office]>
```

Delete the column

14. Empty the table

You can delete all data in a table with the format as below:

```
TRUNCATE table_name;
```

You can see the options for this command on this page. For example, if you want to delete all the data in the employee table, then type the command below:

```
TRUNCATE employee;
```

```
MariaDB [db_office]> SELECT * FROM employee;
+------+------+----------+
| name | age  | city     |
+------+------+----------+
| bob  |   23 | New York |
+------+------+----------+
1 row in set (0.004 sec)

MariaDB [db_office]> TRUNCATE employee;
Query OK, 0 rows affected (0.073 sec)

MariaDB [db_office]> SELECT * FROM employee;
Empty set (0.007 sec)
```

Truncate the table

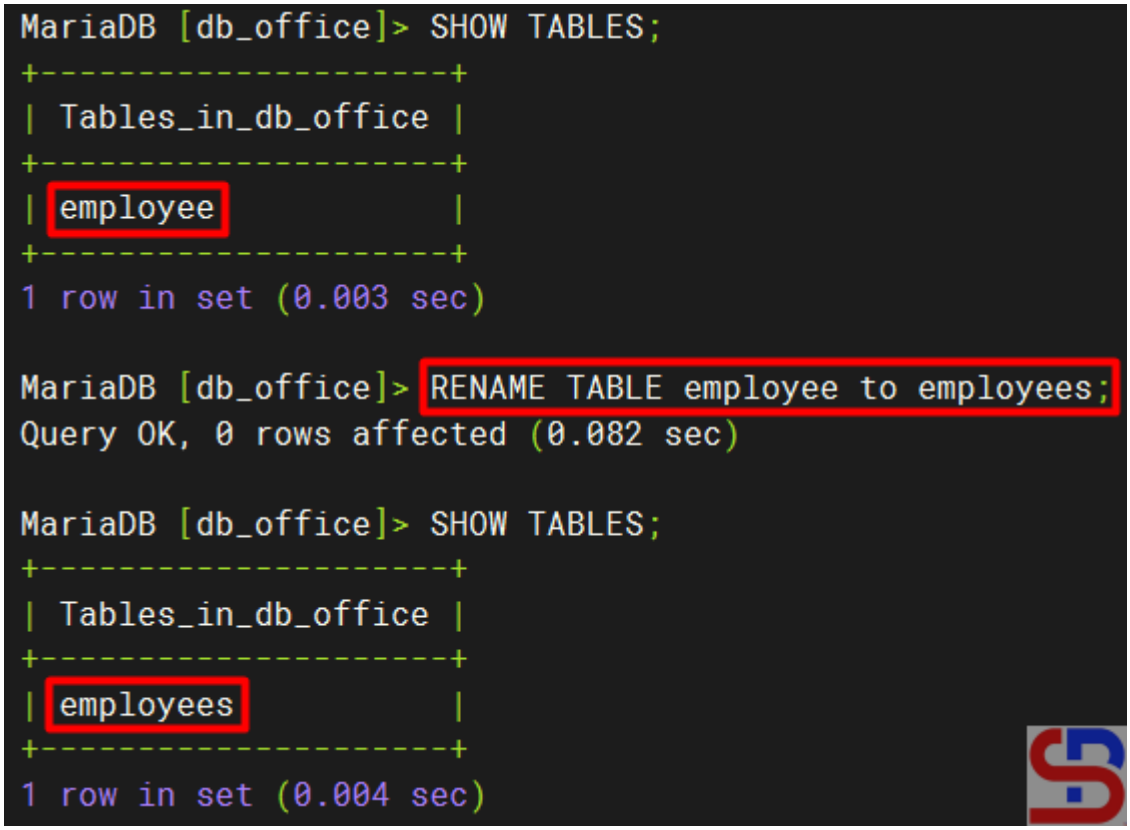
15. Change a table name

You can change the name of the table using the format below:

```
RENAME TABLE old_table_name TO new_table_name;
```

You can see the options for this command on this page. Use the command below if, for example, you want to change the name of the employee table to employees:

```
RENAME TABLE employee to employees;
```



Rename the table

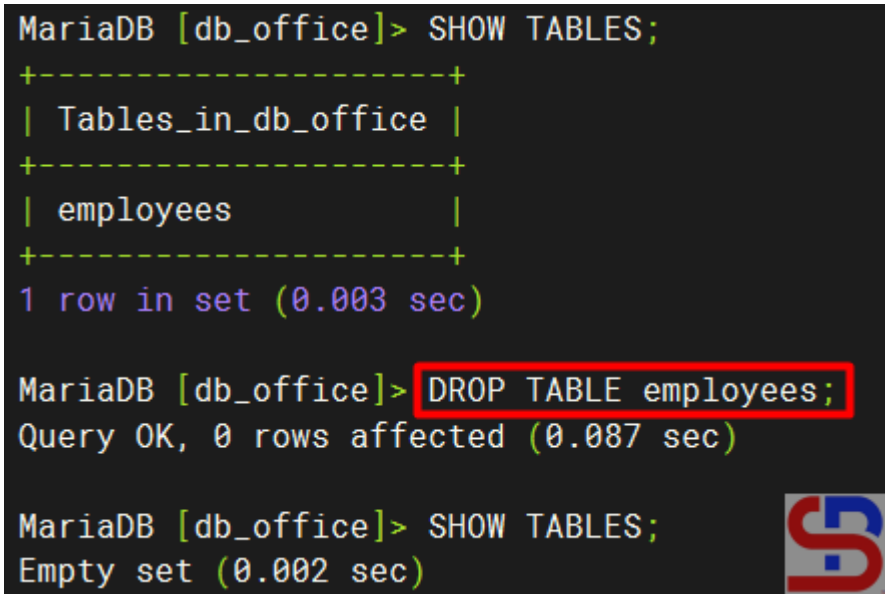
16. Delete a table

Use the format below to delete a table:

```
DROP TABLE table_name;
```

You can see the options for this command on this page. For example, if you want to delete the employee table, then use the command below:

```
DROP TABLE employees;
```



Delete the table

17. Delete a database

To delete a database, use the format below:

```
DROP DATABASE database_name;
```

You can see the options for this command on this page. For example, if you want to delete the db_office database, then type the command below:

```
DROP DATABASE db_office;
```

```
MariaDB [db_office]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| db_office          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.006 sec)

MariaDB [db_office]> DROP DATABASE db_office;
Query OK, 0 rows affected (0.072 sec)

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.002 sec)
```

Delete the database

18. Quit from the database

To quit from the database, run the command below:

```
\q
```

```
MariaDB [(none)]> \q
Bye
sysadmin@docker:~$
```

Quit from the database

Note

The next article will explain how to access a database using a user.

References

MariaDB.com
bertvv.github.io
zuar.com
educba.com
gist.github.com

How to Install the Latest Version of MariaDB on the Linux Server?

written by sysadmin | 17 September 2025

MariaDB is one of the widely used open-source database applications that was first released in 2009. This database was named MySQL, but in 2008, Sun Microsystems acquired MySQL, so the MySQL database maker made MariaDB as a free version. But sometimes when you install MariaDB on your Linux server, your MariaDB version is not the latest version.

Problem

How to install the latest version of MariaDB on the Linux server?

Solution

As of this writing, the latest MariaDB version is 11.7.2. Use the command below to create a MariaDB repository on your Linux server:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash
```

and the process will occur as shown image below:

```
[root@RockyLinux9 ~]# curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash
# [info] Checking for script prerequisites.
# [info] MariaDB Server version 11.rolling is valid
# [info] Repository file successfully written to /etc/yum.repos.d/mariadb.repo
# [info] Adding trusted package signing keys...
/etc/pki/rpm-gpg ~
~
# [info] Successfully added trusted package signing keys
# [info] Cleaning package cache...
73 files removed
[root@RockyLinux9 ~]#
```

Running the command

After that, install MariaDB based on the Linux server distribution that you use:

Redhat-Based Version (Centos, Almalinux, Rockylinux)

```
yum install MariaDB-server MariaDB-client MariaDB-backup
```

Ubuntu/Debian

```
sudo apt-get install mariadb-server mariadb-client mariadb-backup
```

OpenSUSE

```
sudo zypper install MariaDB-server MariaDB-client MariaDB-backup
```

After MariaDB has been installed on your Linux server, use the command below to see the MariaDB version that you have installed:

```
mysql -V
```

```
[root@RockyLinux9 ~]# mysql -V
mysql: Deprecated program name. It will be removed in a future release, use '/usr/bin/mariadb' instead
mysql from 11.7.2-MariaDB, client 15.2 for Linux (x86_64) using  EditLine wrapper
[root@RockyLinux9 ~]#
```

Checking the MariaDB version

After that, use the command below to see the status of MariaDB:

```
systemctl status mariadb
```

```
sysadmin@ubuntu2404:~$ systemctl status mariadb
● mariadb.service - MariaDB 11.7.2 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
    Drop-In: /etc/systemd/system/mariadb.service.d
             └─migrated-from-my.cnf-settings.conf
     Active: active (running) since Sat 2025-04-19 09:42:52 UTC; 1min 0s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 2604 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
    Process: 2606 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`/usr/bin/galera_recovery`; [ $? -eq 0 ]   &&
    Process: 2648 ExecStartPost=/bin/rm -f /run/mysqld/wsrep-start-position (code=exited, status=0/SUCCESS)
    Process: 2650 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 2629 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 7054)
     Memory: 174.9M (peak: 264.2M)
        CPU: 24.956s
     CGroup: /system.slice/mariadb.service
             └─2629 /usr/sbin/mariadbd

Apr 19 09:42:40 ubuntu2404 mariadbd[2629]: 2025-04-19  9:42:40 0 [Note] Plugin 'wsrep-provider' is disabled.
Apr 19 09:42:40 ubuntu2404 mariadbd[2629]: 2025-04-19  9:42:40 0 [Note] InnoDB: Buffer pool(s) load completed at 250419  9:42:40
Apr 19 09:42:51 ubuntu2404 mariadbd[2629]: 2025-04-19  9:42:51 0 [Note] Server socket created on IP: '127.0.0.1'.
Apr 19 09:42:51 ubuntu2404 mariadbd[2629]: 2025-04-19  9:42:51 0 [Note] mariadbd: Event Scheduler: Loaded 0 events
Apr 19 09:42:51 ubuntu2404 mariadbd[2629]: 2025-04-19  9:42:51 0 [Note] /usr/sbin/mariadbd: ready for connections.
Apr 19 09:42:51 ubuntu2404 mariadbd[2629]: Version: '11.7.2-MariaDB-ubu2404'  socket: '/run/mysqld/mysqld.sock'  port: 3306  mariadb.org b
Apr 19 09:42:52 ubuntu2404 systemd[1]: Started mariadb.service - MariaDB 11.7.2 database server.
Apr 19 09:42:52 ubuntu2404 /etc/mysql/debian-start[2652]: Upgrading MariaDB tables if necessary.
Apr 19 09:42:53 ubuntu2404 /etc/mysql/debian-start[2665]: Checking for insecure root accounts.
Apr 19 09:42:54 ubuntu2404 /etc/mysql/debian-start[2669]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Ar
sysadmin@ubuntu2404:~$
```

Display the status of service

You can see the image above that MariaDB's service is on. However, if MariaDB's service is still not on, use the command below to turn on the MariaDB service:

```
systemctl start mariadb
```

To run MariaDB after turning on the server, use the command below:

```
systemctl enable mariadb
```

```
sysadmin@ubuntu2404:~$ sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mariadb
sysadmin@ubuntu2404:~$
```

Enable MariaDB

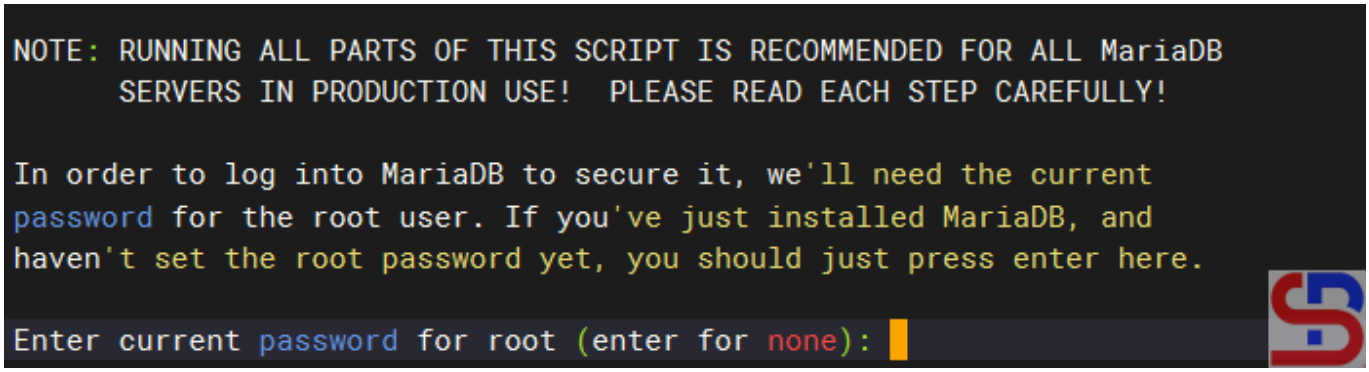
After that, to MariaDB becomes safe, use one of the commands below:

```
sudo mariadb_secure_installation
```

OR

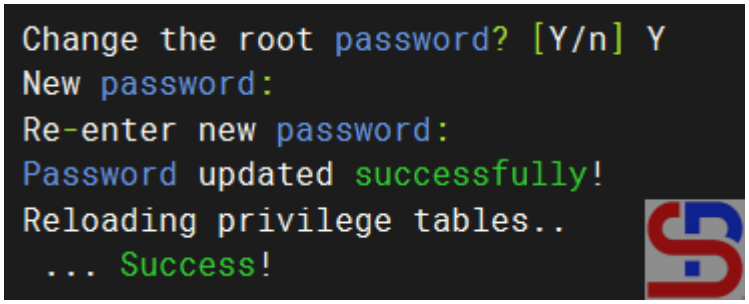
```
sudo mariadb-secure-installation
```

Then there will be a display as in the image below:



Running the command

By default, there is no password when accessing MariaDB so press the **Enter** button to continue the process. After that, you must answer the questions displayed, including creating a new password as shown below:



Write the password

Then, continue until the process is finished. The next article will explain how to manage a database and its table(s) in MariaDB.

Note

Use the command below if you want to see the option when you run the command to make a repo:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash -s -- --help
```

If you want to install a certain version of MariaDB, for

example, version 11.4, then you can use the command below:

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash -s -
- --mariadb-server-version="mariadb-11.4"
```

If you want to download a script, use the command below:

```
curl -LO https://r.mariadb.com/downloads/mariadb_repo_setup
```

References

mariadb.com
devopscube.com