How To Create a New User and Grant Permissions in MySQL

150

Posted June 12, 2012 3.8m MYSQL

By: Etel Sverdlov

Introduction

MySQL is an open-source database management software that helps users store, organize, and later retrieve data. It has a variety of options to grant specific users nuanced permissions within the tables and databases—this tutorial will give a short overview of a few of the many options.

SCROLL TO TOP

What the Highlights Mean

Throughout this tutorial, any lines that the user needs to enter or customize will be highlighted! The rest should mostly be copy-and-pastable.

How to Create a New User

In Part 1 of the MySQL Tutorial, we did all of the editing in MySQL as the root user, with full access to all of the databases. However, in cases where more restrictions may be required, there are ways to create users with custom permissions.

Let's start by making a new user within the MySQL shell:

```
mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

> **Note:** When adding users within the MySQL shell in this tutorial, we will specify the user's host as `localhost` and not the server's IP address. `localhost` is a hostname which means "this computer," and MySQL treats this particular hostname specially: when a user with that host logs into MySQL it will attempt to connect to the local server by using a Unix socket file. Thus, `localhost` is typically used when you plan to connect by SSHing into your server or when you're running the local `mysql` client to connect to the local MySQL server.

At this point newuser has no permissions to do anything with the databases. In fact, even if newuser tries to login (with the password, password), they will not be able to reach the MySQL shell.

Therefore, the first thing to do is to provide the user with access to the information they will need.

```
mysql> GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

The asterisks in this command refer to the database and table (respectively) that they can access—this specific command allows to the user to read, edit, execute and perform all tasks across all the databases and tables.

Please note that in this example we are granting newuser full root access to everything in our database. While this is helpful for explaining some MySQL concepts, it may be impractical for most use cases and could put your database's security at high risk.

Once you have finalized the permissions that you want to set up for your new users, always be sure to reload all the privileges.

```
mysql> FLUSH PRIVILEGES;
```

Your changes will now be in effect.

How To Grant Different User Permissions

Here is a short list of other common possible permissions that users can enjoy.

- ALL PRIVILEGES- as we saw previously, this would allow a MySQL user full access to a designated database (or if no database is selected, global access across the system)
- CREATE- allows them to create new tables or databases
- DROP- allows them to them to delete tables or databases
- DELETE- allows them to delete rows from tables
- INSERT- allows them to insert rows into tables
- SELECT- allows them to use the `SELECT` command to read through databases
- UPDATE- allow them to update table rows
- GRANT OPTION- allows them to grant or remove other users' privileges

To provide a specific user with a permission, you can use this framework:

```
mysql> GRANT type_of_permission ON database_name.table_name TO ‘username’@'localhost’;
```

If you want to give them access to any database or to any table, make sure to put an asterisk (*) in the place of the database name or table name.

Each time you update or change a permission be sure to use the Flush Privileges command.

If you need to revoke a permission, the structure is almost identical to granting it:

```
mysql> REVOKE type_of_permission ON database_name.table_name FROM ‘username’@‘localhost’
```

Note that when revoking permissions, the syntax requires that you use `FROM`, instead of `TO` as we used when granting permissions.

You can review a user’s current permissions by running the following:

```
SHOW GRANTS username;
```

Just as you can delete databases with DROP, you can use DROP to delete a user altogether:

```
mysql> DROP USER ‘username’@‘localhost’;
```

To test out your new user, log out by typing:

```
mysql> quit
```

and log back in with this command in terminal:

```
$ mysql -u [username] -p
```

Conclusion

After completing this tutorial, you should have a sense of how to add new users and grant them a variety of permissions in a MySQL database. From here, you could continue to explore and experiment with different permissions settings for your database, or you may want to learn more about some higher-level MySQL configurations.

For more information about the basics of MySQL, we encourage you to check out the following tutorials:

- How To Import and Export Databases and Reset a Root Password in MySQL
- How To Migrate a MySQL Database Between Two Servers
- How To Set Up Master Slave Replication in MySQL

By: Etel Sverdlov

♡ Upvote (150) Subscribe Share

Related Tutorials

How To Set Up a Remote Database to Optimize Site Performance with MySQL

How To Migrate a MySQL Database Between Two Servers

How To Set Up Master Slave Replication in MySQL

How To Import and Export Databases and Reset a Root Password in MySQL

How To Set Up Laravel, Nginx, and MySQL with Docker Compose

48 Comments

Leave a comment...

Log In to Comment

chris96043 *February 13, 2013*

1 You can grant multiple privileges in one command by separating them with commas:
eg: "GRANT UPDATE, SELECT ON [database name].[table name] TO ‘[username]’@‘localhost’;"

f.bagnardi *March 6, 2013*

0 This code from above has a backtick before localhost. It should be a single quote.

GRANT ALL PRIVILEGES ON * . * TO 'newuser'@‘localhost';

etel *March 6, 2013*

0 Updated

digitalocean114189 *March 12, 2013*

0 Hi, nice intro. It was useful.
I noticed that for the REVOKE command, one has to use FROM, not TO.
Also, might be helpful for new users to know that they can use '%' as a wildcard instead of 'localhost'.

neilh20 *March 13, 2013*

0 This worked for me. However to be able to use MySql Workbench it seems it wants another version of the user.
I needed to do the following (which has taken me a few hours of playing around with to get right)
mysql> select user,host from mysql.user; GRANT ALL ON *.* to user@’%’ IDENTIFIED BY 'user-pwd';
mysql> FLUSH PRIVILEGES;
mysql> select user,host from mysql.user;
mysql>quit
Also need to comment out or change the bind-address to . This does reduce security.
sudo nano /etc/mysql/my.cnf
;bind-address=127.0.0.1
exit and

```
$service mysql start
$service mysql stop
```

then get access on from my Sql workbench using user/user-pwd on std port for adminstering, creating and querying.

ClockworkHero *March 31, 2013*

0 There's no link at the start of this tutorial to the first tutorial. Can you please add that link?

---

nulikiran4u *May 23, 2013*

0 useful

---

ravuri.srinivasarao7 *June 28, 2013*

0 how to give permission to only select views in requried user pls send me urgent

---

kamaln7 MOD *June 28, 2013*

0 @ravuri.srinivasarao7: Please read the second part of the article: "How To Grant Different User Permissions"

---

bluethrustweb *July 28, 2013*

0 Is there a way to just give permission to create a new table within the specified database, but not allow the creation of a new database?

---

kamaln7 MOD *July 28, 2013*

0 @bluethrustweb: Yes, of course:

```
GRANT CREATE ON `database` TO 'user'@'host';
```

---

mzengaekamkulu *August 2, 2013*

0 Am a new comer in mysql server, I don't even why are creating these users and grant to them privileges! may i have some description plz?

kamaln7 MOD *August 2, 2013*

0 @mzengaekamkulu: What do you mean? You have to create a user for each app you use so it can connect to the mysql server.

bda.awk *October 8, 2013*

0 How to configure remote access to my mysql?

info268212 *October 11, 2013*

0 where the *heck* is Part 1 of the Tut?

kamaln7 MOD *October 12, 2013*

0 @info: Part 1: https://www.digitalocean.com/community/articles/a-basic-mysql-tutorial

A Basic MySQL Tutorial

by Etel Sverdlov

MySQL is a powerful database management system used for organizing and retrieving data. This tutorial explains how to access the MySQL Shell, how to create and delete a new database, how to create and delete a new table, and how to add, delete, and update columns mario.medarz *October 19, 2013*

0 Very easy and clear! Thanks for these worthy and handy tutorials!

---

fernandoaleman *October 21, 2013*

0 You have an error in your REVOKE command. Instead of:

REVOKE [type of permission] ON [database name].[table name] TO ‘[username]’@‘localhost’;

it should be:

REVOKE [type of permission] ON [database name].[table name] FROM ‘[username]’@‘localhost’;

---

kamaln7 MOD *October 22, 2013*

0 @fernandoaleman: Thanks! I've correct the article.

---

petre.octavian *December 12, 2013*

1 Hi,

I have done all the above but I can't seem to log with the new user with:

mysql -u [username]-p

I have the error:

ERROR 1045 (28000): Access denied for user 'prosper202'@'localhost' (using password: YES)

What may be the problem?

Load More Comments