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

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

Sadly, at this point newuser has no permissions to do anything with the databases. In fact, if newuser even 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.

```
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.

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

```
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 all access to a designated database (or if no database is selected, 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:

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:

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

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

DROP USER 'demo'@'localhost';

To test out your new user, log out by typing

quit

and log back in with this command in terminal:

mysql -u [username]-p # # Connect to the local database server as user root # You will be

prompted for a password. # mysql -h localhost -u root -p # # Now we see the 'mysql>' prompt and we can run # the following to create a new database for Paul. # mysql> create database pauldb; Query OK, 1 row affected (0.00 sec) # # Now we create the user paul and give him full # permissions on the new database mysql> grant CREATE,INSERT,DELETE,UPDATE,SELECT on pauldb.* to paul@localhost; Query OK, 0 rows affected (0.00 sec) # # Next we set a password for this new user # mysql> set password for paul = password('mysecretpassword'); Query OK, 0 rows affected (0.00 sec) # # Cleanup and ext mysql> flush privileges; mysql> exit;
Â
If you use

mysql> grant CREATE,INSERT,DELETE,UPDATE,SELECT on pauldb.* to paul@localhost identified by 'secretpassword';

Then you don't need to flush the grant tables.

or

mysql -h localhost -u root -p
mysql> create database paul;
mysql> grant ALL on paul.* to paul@localhost;
mysql> set password for paul = password('foobar');
1) This will create a databse named paul
2) This grants ALL (all normal stuff) for paul in his database called paul.
3) This will set paul's password to foobar.
mysql> exit;