|  |  |
| --- | --- |
|  |  |
|  |

|  |
| --- |
| **Lab 06: Working with USERs and PRIVILEGEs** |

The ability to create users and manage privileges provides a granular level of security and access in MySQL. In this lab, you will be tasked with creating new users in the MySQL server and deleting users that no longer need access. Once the users are created, you will need to grant the appropriate privileges to each user so that they have access to the correct databases and tables within the server.

**Objective(s):**

1. Create Users on the MySQL Server
2. Remove User(s) and Modify Users
3. Granting due privileges to users
4. Revoking privileges from user

|  |
| --- |
| **1: Introduction of USERs** |

Upon installation, MySQL creates a root user account which you can use to manage your database. This user has full privileges over the MySQL server, meaning it has complete control over every database, table, user, and so on. Because of this, it’s best to avoid using this account outside of administrative functions. This step outlines how to use the root MySQL user to create a new user account and grant it privileges. Once you have access to the MySQL prompt or Workbench, you can create a new user with a CREATE USER statement. These follow this general syntax.

**Syntax:**

|  |
| --- |
| CREATE USER 'username'@'host' IDENTIFIED WITH authentication\_plugin BY 'password'; |

After CREATE USER, you specify a username. This is immediately followed by @ sign and then the hostname from which this user will connect. If you only plan to access this user locally from your Ubuntu server, you can specify localhost. Wrapping both the username and host in single quotes isn’t always necessary, but doing so can help to prevent errors.

You have several options when it comes to choosing your user’s authentication plugin. The auth\_socket plugin mentioned previously can be convenient, as it provides strong security without requiring valid users to enter a password to access the database. But it also prevents remote connections, which can complicate things when external programs need to interact with MySQL.

As an alternative, you can leave out the WITH authentication\_plugin portion of the syntax entirely to have the user authenticate with MySQL’s default plugin, caching\_sha2\_password. Some authentication plugins store account credentials internally to MySQL, in the mysql.user system table:

1. mysql\_native\_password
2. caching\_sha2\_password
3. sha256\_password

Run the following command to create a user that authenticates with caching\_sha2\_password. Be sure to change sammy to your preferred username and password to a strong password of your choosing:

**Example 1:**

|  |
| --- |
| CREATE USER 'sammy'@'localhost' IDENTIFIED BY 'password'; |

After a user is created you can make changes in password by using ALTER USER command.

**Syntax:**

|  |
| --- |
| ALTER USER 'username'@'hostname' IDENTIFIED WITH authentication\_plugin BY 'new\_password'; |

**Example 2:**

|  |
| --- |
| ALTER USER 'sammy'@'localhost' IDENTIFIED WITH caching\_sha2\_password BY 'password'; |

**Example 3:**

|  |
| --- |
| ALTER USER 'sammy'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password'; |

**Example 4:**

|  |
| --- |
| ALTER USER USER() IDENTIFIED BY 'password'; |

This will change the password of user currently logged in.

|  |
| --- |
| **2: Removing Users** |

A user can be removed (just like tables) using drop command.

**Syntax:**

|  |
| --- |
| DROP USER 'username'@'hostname'; |

**Example 5:**

|  |
| --- |
| DROP USER 'sammy'@'localhost'; |

|  |
| --- |
| **3: User PREVILEGEs (Permissions)** |

The privileges granted to a MySQL account determine which operations the account can perform. MySQL privileges differ in the contexts in which they apply and at different levels of operation:

1. Administrative privileges enable users to manage operation of the MySQL server. These privileges are global because they are not specific to a particular database.
2. Database privileges apply to a database and to all objects within it. These privileges can be granted for specific databases, or globally so that they apply to all databases.
3. Privileges for database objects such as tables, indexes, views, and stored routines can be granted for specific objects within a database, for all objects of a given type within a database (for example, all tables in a database), or globally for all objects of a given type in all databases.

Privileges also differ in terms of whether they are static (built in to the server) or dynamic (defined at runtime). Whether a privilege is static or dynamic affects its availability to be granted to user accounts and roles.

|  |
| --- |
| **4: Summary / Types of Privileges** |

MySQL supports static and dynamic privileges:

1. **Static privileges** are built in to the server. They are always available to be granted to user accounts and cannot be unregistered.
2. **Dynamic privileges** can be registered and unregistered at runtime. This affects their availability: A dynamic privilege that has not been registered cannot be granted.

For example, the SELECT and INSERT privileges are static and always available, whereas a dynamic privilege becomes available only if the component that implements it has been enabled.

Summary of Available Privileges

The following table shows the static privilege names used in GRANT and REVOKE statements, along with the column name associated with each privilege in the grant tables and the context in which the privilege applies.

|  |  |  |
| --- | --- | --- |
| **Privilege** | **Grant Table Column** | **Context** |
| [ALL [PRIVILEGES]](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_all) | Synonym for “all privileges” | Server administration |
| [ALTER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_alter) | Alter\_priv | Tables |
| [ALTER ROUTINE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_alter-routine) | Alter\_routine\_priv | Stored routines |
| [CREATE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create) | Create\_priv | Databases, tables, or indexes |
| [CREATE ROLE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-role) | Create\_role\_priv | Server administration |
| [CREATE ROUTINE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-routine) | Create\_routine\_priv | Stored routines |
| [CREATE TABLESPACE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-tablespace) | Create\_tablespace\_priv | Server administration |
| [CREATE TEMPORARY TABLES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-temporary-tables) | Create\_tmp\_table\_priv | Tables |
| [CREATE USER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-user) | Create\_user\_priv | Server administration |
| [CREATE VIEW](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-view) | Create\_view\_priv | Views |
| [DELETE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_delete) | Delete\_priv | Tables |
| [DROP](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_drop) | Drop\_priv | Databases, tables, or views |
| [DROP ROLE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_drop-role) | Drop\_role\_priv | Server administration |
| [EVENT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_event) | Event\_priv | Databases |
| [EXECUTE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_execute) | Execute\_priv | Stored routines |
| [FILE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_file) | File\_priv | File access on server host |
| [GRANT OPTION](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_grant-option) | Grant\_priv | Databases, tables, or stored routines |
| [INDEX](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_index) | Index\_priv | Tables |
| [INSERT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_insert) | Insert\_priv | Tables or columns |
| [LOCK TABLES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_lock-tables) | Lock\_tables\_priv | Databases |
| [PROCESS](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_process) | Process\_priv | Server administration |
| [PROXY](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_proxy) | See proxies\_priv table | Server administration |
| [REFERENCES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_references) | References\_priv | Databases or tables |
| [RELOAD](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_reload) | Reload\_priv | Server administration |
| [REPLICATION CLIENT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_replication-client) | Repl\_client\_priv | Server administration |
| [REPLICATION SLAVE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_replication-slave) | Repl\_slave\_priv | Server administration |
| [SELECT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_select) | Select\_priv | Tables or columns |
| [SHOW DATABASES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_show-databases) | Show\_db\_priv | Server administration |
| [SHOW VIEW](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_show-view) | Show\_view\_priv | Views |
| [SHUTDOWN](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_shutdown) | Shutdown\_priv | Server administration |
| [SUPER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_super) | Super\_priv | Server administration |
| [TRIGGER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_trigger) | Trigger\_priv | Tables |
| [UPDATE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_update) | Update\_priv | Tables or columns |
| [USAGE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_usage) | Synonym for “no privileges” | Server administration |

For the dynamic privilege names used in GRANT and REVOKE statements, along with the context in which the privilege applies, refer to [Official MySQL Documentation](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html).

|  |
| --- |
| **5: GRANT Privileges to USER** |

**Syntax:**

|  |
| --- |
| GRANT **PRIVILEGE** ON database.table TO 'username'@'host'; |

The PRIVILEGE value in this example syntax defines what actions the user is allowed to perform on the specified database and table. You can grant multiple privileges to the same user in one command by separating each with a comma. You can also grant a user privileges globally by entering asterisks (\*) in place of the database and table names. In SQL, asterisks are special characters used to represent “all” databases or tables.

|  |
| --- |
| WARNING: Such broad privileges **should not be granted lightly**, as anyone with access to this MySQL user will have complete control over every database on the server. |

Some users may want to grant their MySQL user the ALL PRIVILEGES privilege, which will provide them with broad super-user privileges akin to the root user’s privileges, like so:

**Example 6:**

|  |
| --- |
| GRANT ALL PRIVILEGES ON \*.\* TO 'sammy'@'localhost' WITH GRANT OPTION; |

**Example 7:**

Grant all privileges on the dev database to the mike and corey user:

|  |
| --- |
| GRANT ALL ON dev.\* TO 'corey'@'localhost'; |

**Example 8:**

Grant INSERT and SELECT privileges on the products table in the dev database to will:

|  |
| --- |
| GRANT **SELECT, INSERT** ON dev.products TO 'will'@'localhost'; |

**Example 9:**

Grant INSERT and SELECT privileges on the products table in the dev database to will:

|  |
| --- |
| GRANT **SELECT, INSERT** ON dev.products TO 'will'@'localhost'; |

**Example 10:**

|  |
| --- |
| GRANT ***CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD*** on \*.\* TO 'sammy'@'localhost' WITH GRANT OPTION; |

To illustrate, the following command grants a user global privileges to CREATE, ALTER, and DROP databases, tables, and users, as well as the power to INSERT, UPDATE, and DELETE data from any table on the server. It also grants the user the ability to query data with SELECT, create foreign keys with the REFERENCES keyword, and perform FLUSH operations with the RELOAD privilege. However, you should only grant users the permissions they need, so feel free to adjust your own user’s privileges as necessary.

|  |
| --- |
| **6: REVOKE Privileges from USER** |

Many guides suggest running the FLUSH PRIVILEGES command immediately after a CREATE USER or GRANT statement in order to reload the grant tables to ensure that the new privileges are put into effect:

**Example 11:**

|  |
| --- |
| FLUSH PRIVILEGES; |

However, according to the official MySQL documentation, when you modify the grant tables indirectly with an account management statement like GRANT, the database will reload the grant tables immediately into memory, meaning that the FLUSH PRIVILEGES command isn’t necessary in our case. On the other hand, running it won’t have any negative effect on the system.

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

**Syntax:**

|  |
| --- |
| REVOKE **TYPE\_ OF\_PRIVILEGE** ON database.table FROM 'username'@'host'; |

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

**Example 12:**

Revoke the INSERT privilege from the kenny user running the following command:

|  |
| --- |
| REVOKE **INSERT** ON prod.orders FROM 'kenny'@'host'; |

**Lab Task(s):**

Exercise

1. Create the Following Four Users on the MySQL Server: `corey`, `will`, `mike`, and `myles`, and Allow Logins from `localhost` Only, Using the Password `Linux4you!`.
2. Hit and Run all the given queries in the Lab Manual
3. Create the Following Four Users on the MySQL Server: `Admin`, `Student`, `Faculty` and Guest. Allow Logins from `localhost` Only, Using the Password `SukkurIBA`
4. Create an Attendance database management System in such a way:
   1. Admin can create database, users, tables with grant option
   2. Faculty can Input and Modify Attendance
   3. Students can only view Attendance

* Table Attendance (Teacher ID, Student ID, Course ID, Date, Attendance: P/A )
* Table Faculty (ID, Name)
* Table Student (ID, Name)
* Table Course (ID, Name)

|  |
| --- |
| **Use Dummy Data and Assume Data values in case of Missing Information** |

**END**