Name: Saurabh Kadam

Roll No.: 47

Course: PGDACC (ESDS)

Subject: DBA

Assignment No.4

**Title**: Understanding Users and their Privileges

**Aim**: To perform different kind of DB user related operations

* New user creation with password
* Assigning all privileges to newly created user
* Assigning read only privileges to newly created user
* Assigning write only privileges to newly created user
* Changing password of previously created user
* Checking read only and write only users work like permission you assigned to them
* Revoking all permissions of user previously created
* Deletion of previously created user

**Theory:**

# **SQL GRANT REVOKE Commands**

# DCL commands are used to enforce database security in a multiple user database environment. Two types of DCL commands are GRANT and REVOKE. Only Database Administrator's or owner's of the database object can provide/remove privileges on a database object.

1. **SQL GRANT Command**

SQL GRANT is a command used to provide access or privileges on the database objects to the users.

**The Syntax for the GRANT command is:**

*GRANT privilege\_name*

*ON object\_name*

*TO {user\_name |PUBLIC |role\_name}*

*[WITH GRANT OPTION];*

* ***privilege\_name*** is the access right or privilege granted to the user. Some of the access rights are ALL, EXECUTE, and SELECT.
* ***object\_name*** is the name of a database object like TABLE, VIEW, STORED PROC and SEQUENCE.
* ***user\_name*** is the name of the user to whom an access right is being granted.
* ***user\_name*** is the name of the user to whom an access right is being granted.
* ***PUBLIC*** is used to grant access rights to all users.
* ***ROLES*** are a set of privileges grouped together.
* ***WITH GRANT OPTION*** - allows a user to grant access rights to other users.

1. **SQL REVOKE Command:**

The REVOKE command removes user access rights or privileges to the database objects.

The Syntax for the REVOKE command is:

*REVOKE privilege\_name*

*ON object\_name*

*FROM {user\_name |PUBLIC |role\_name}*

1. **Privileges:**

Privileges: Privileges defines the access rights provided to a user on a database object. There are two types of privileges.

1. **System privileges** - This allows the user to CREATE, ALTER, or DROP database objects
2. **Object privileges** - This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data from database objects to which the privileges apply.

Few CREATE system privileges are listed below:

|  |  |
| --- | --- |
| **System Privileges** | **Description** |
| CREATE object | allows users to create the specified object in their own schema. |
| CREATE ANY object | allows users to create the specified object in any schema. |

**The above rules also apply for ALTER and DROP system privileges.**

Few of the object privileges are listed below:

|  |  |
| --- | --- |
| **Object Privileges** | **Description** |
| INSERT | allows users to insert rows into a table. |
| SELECT | allows users to select data from a database object. |
| UPDATE | allows user to update data in a table. |
| EXECUTE | allows user to execute a stored procedure or a function. |

1. **Create a new MySQL User Account**

A user account in MySQL consists of a user name and host name parts.

To create a new MySQL user account, run the following command:

* *CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'user\_password';*

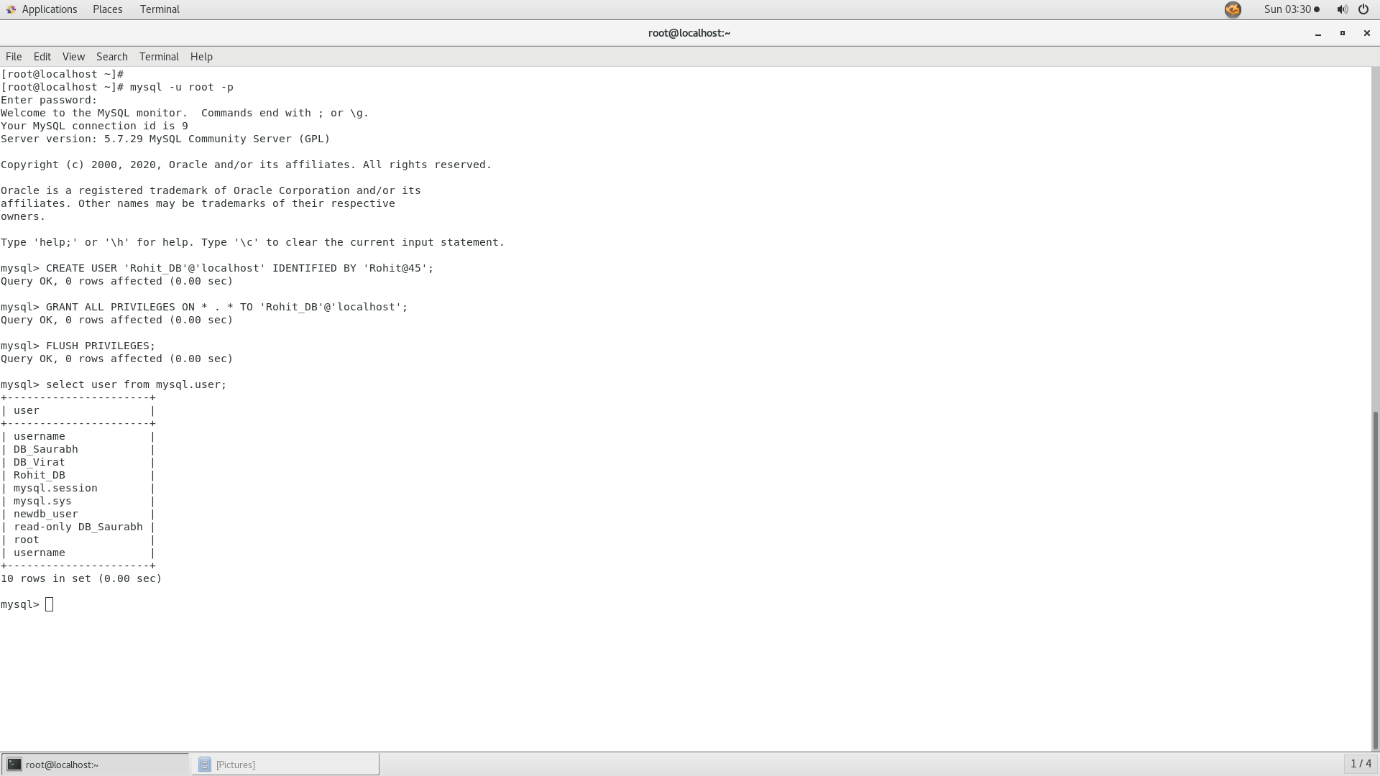
In the command above the hostname part is set to localhost, which means that the user will be able to connect to the MySQL server only from the localhost (i.e from the system where MySQL Server runs).

To grant access from another host, change the hostname part (localhost) with the remote machine IP. For example, to grant access from a machine with IP 10.8.0.5 you would run:

* *CREATE USER 'newuser'@'10.8.0.5' IDENTIFIED BY 'user\_password';*

To create a user that can connect from any host, use the '%' wildcard as a host part:

* *CREATE USER 'newuser'@'%' IDENTIFIED BY 'user\_password';*



1. **Grant Privileges to a MySQL User Account**

There are multiple types of privileges that can be granted to a user account.

The most commonly used privileges are:

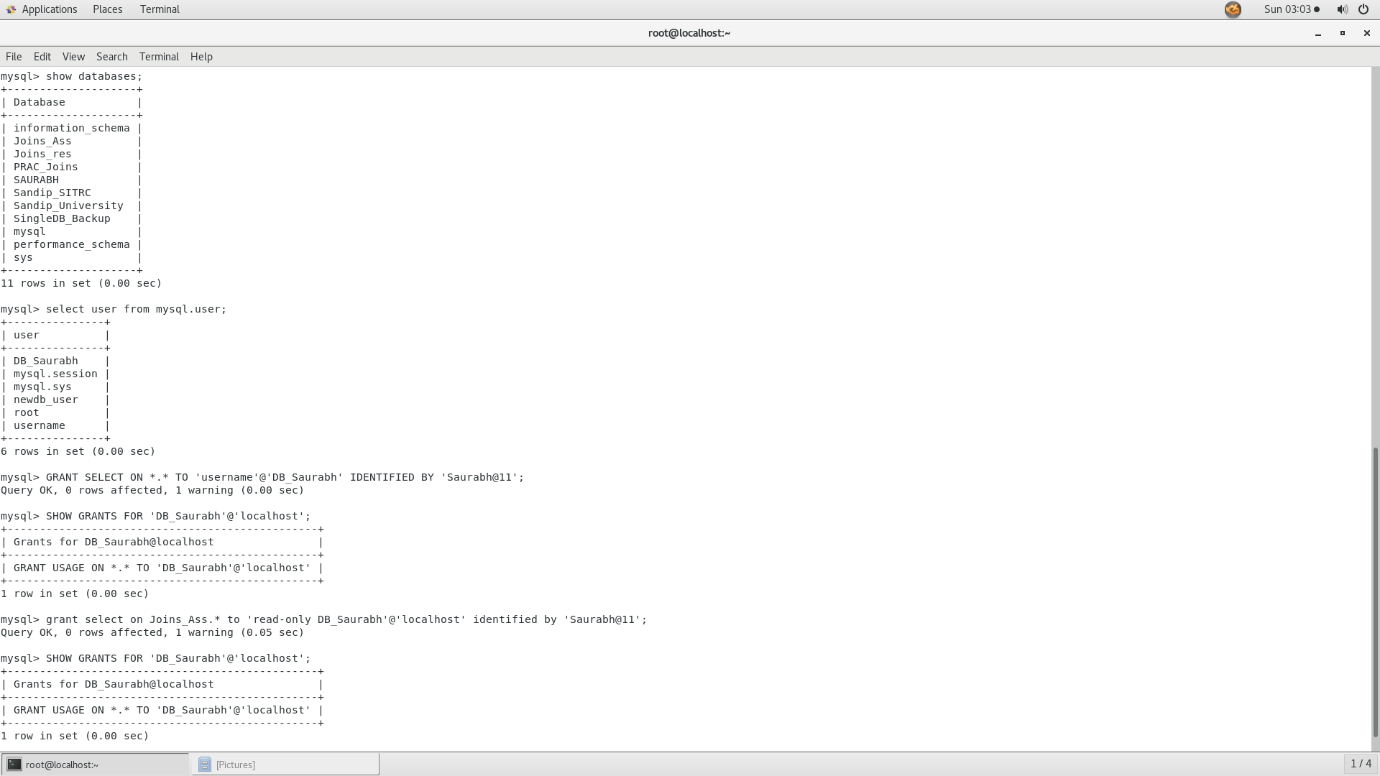
* ALL PRIVILEGES – Grants all privileges to a user account.
* CREATE – The user account is allowed to create databases and tables.
* DROP - The user account is allowed to drop databases and tables.
* DELETE - The user account is allowed to delete rows from a specific table.
* INSERT - The user account is allowed to insert rows into a specific table.
* SELECT – The user account is allowed to read a database.
* UPDATE - The user account is allowed to update table rows.

To grant specific privileges to a user account, you can use the following syntax:

* *GRANT permission1, permission2 ON database\_name.table\_name TO 'database\_user'@'localhost';*

Here are some examples:

* Grand all privileges to a user account over a specific database:
* *GRANT ALL PRIVILEGES ON database\_name.\* TO 'database\_user'@'localhost';*
* Grand all privileges to a user account on all databases:
* *GRANT ALL PRIVILEGES ON \*.\* TO 'database\_user'@'localhost';*
* Grand all privileges to a user account over a specific table from a database:
* *GRANT ALL PRIVILEGES ON database\_name.table\_name TO 'database\_user'@'localhost';*
* Grant multiple privileges to a user account over a specific database:
* *GRANT SELECT, INSERT, DELETE ON database\_name.\* TO database\_user@'localhost';*



**Flush privileges**

* *FLUSH PRIVILEGES;*

when we grant some privileges for a user, running the command flush privileges will reloads the grant tables in the mysql database enabling the changes to take effect without reloading or restarting mysql service.

1. **Display MySQL User Account Privileges**

To find the privilege(s) granted to a specific MySQL user account, use the SHOW GRANTS statement:

* *SHOW GRANTS FOR 'database\_user'@'localhost';*

+---------------------------------------------------------------------------+

| Grants for database\_user@localhost |

+---------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'database\_user'@'localhost' |

| GRANT ALL PRIVILEGES ON `database\_name`.\* TO 'database\_user'@'localhost' |

+---------------------------------------------------------------------------+



1. **Set the MySQL user password**

Type the following commands if you have MySQL 5.7.6 and later or MariaDB 10.1.20 and later:

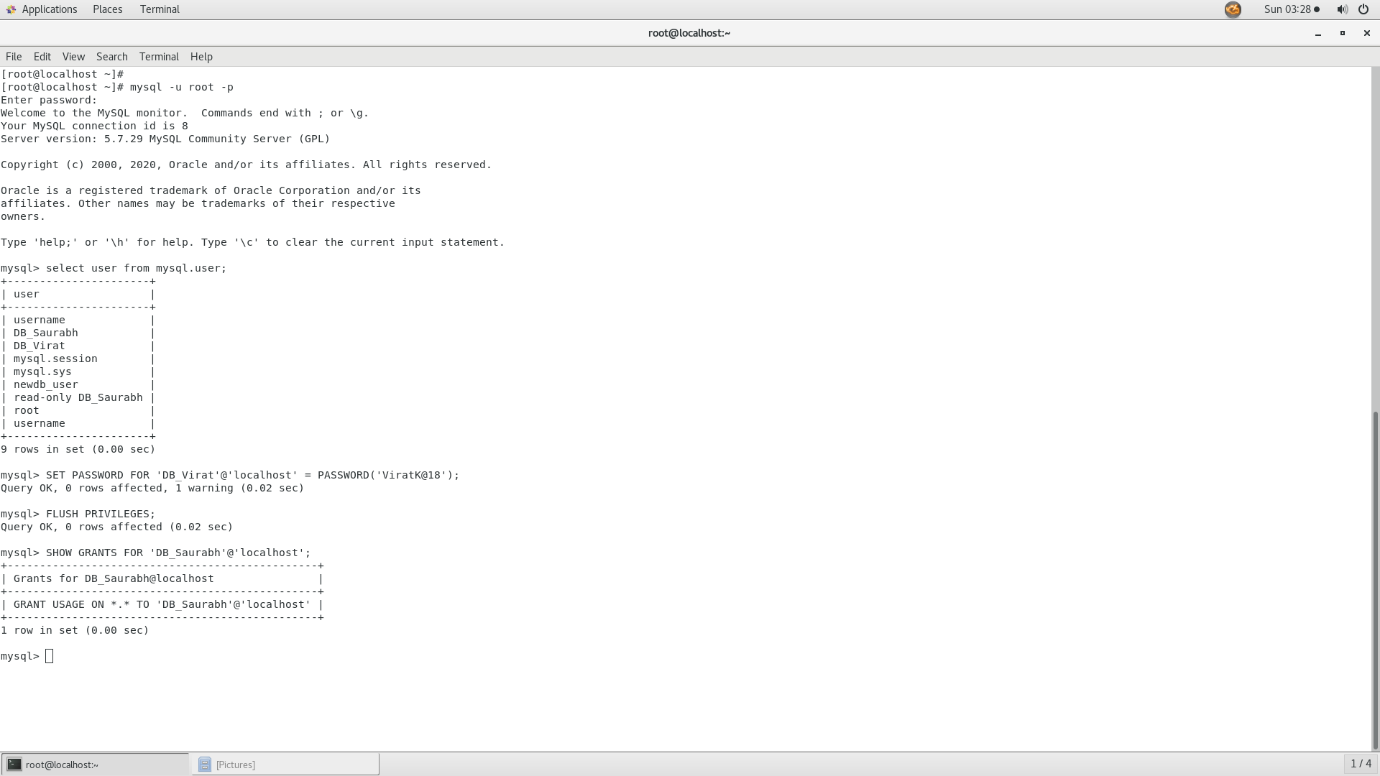
* *ALTER USER 'user-name'@'localhost' IDENTIFIED BY 'NEW\_USER\_PASSWORD';*
* *FLUSH PRIVILEGES;*

If ALTER USER statement doesn’t work for you, you can modify the user table directly:

* *UPDATE mysql.user SET authentication\_string = PASSWORD('NEW\_USER\_PASSWORD') WHERE User = 'user-name' AND Host = 'localhost';*
* *FLUSH PRIVILEGES;*

Type the following commands if you have MySQL 5.7.5 and earlier or MariaDB 10.1.20 and earlier:

* *SET PASSWORD FOR 'user-name'@'localhost' = PASSWORD('NEW\_USER\_PASSWORD');*
* *FLUSH PRIVILEGES;*



1. **Revoke Privileges from a MySQL User Account**

The syntax to revoke one or more privileges from a user account is almost identical as when granting privileges.

For example, to revoke all privileges from a user account over a specific database, use the following command:

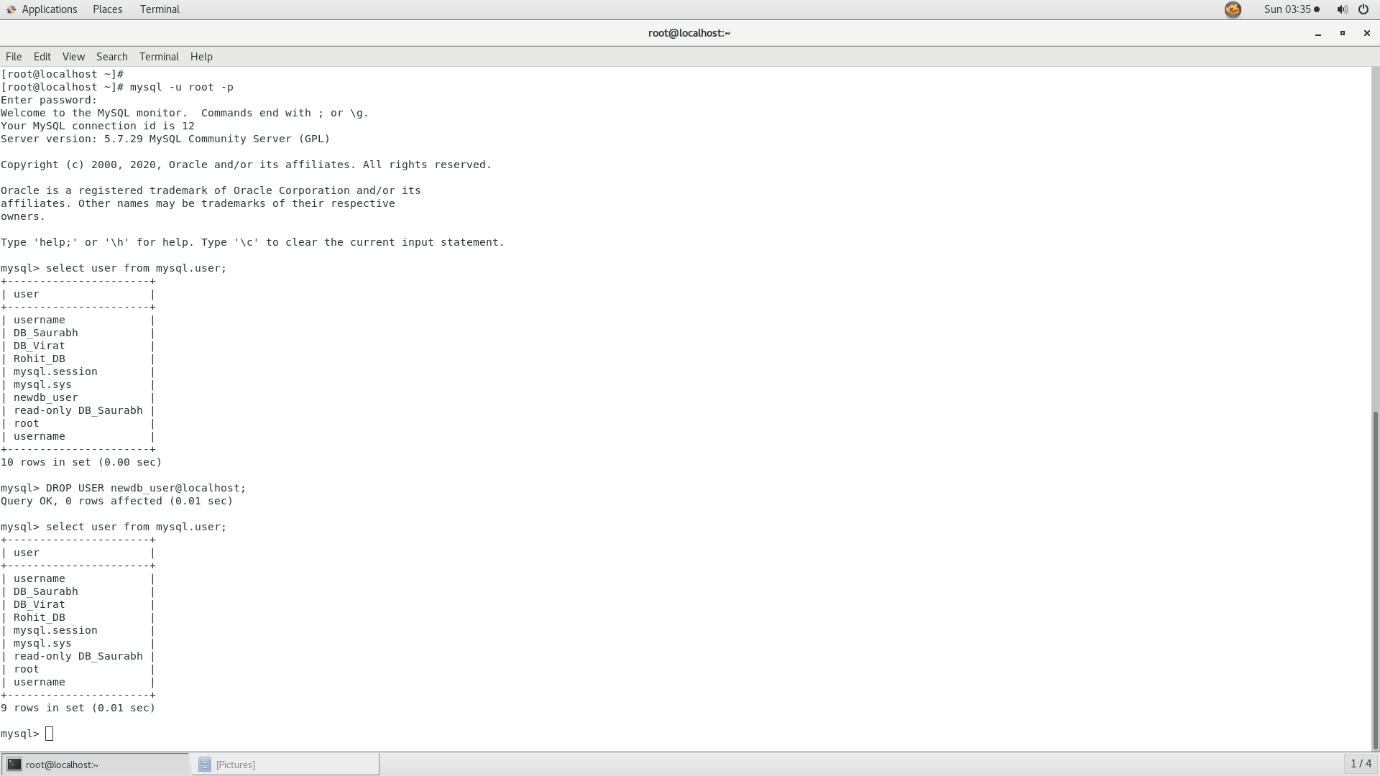
* *REVOKE ALL PRIVILEGES ON database\_name.\* FROM 'database\_user'@'localhost';*

1. **Remove an Existing MySQL User Account**

To delete a MySQL user account, use the DROP USER statement:

* *DROP USER 'user'@'localhost'*

The command above will remove the user account and its privileges.



**Conclusion:**

After completing this practical, we have a sense of how to add new users, grant them a variety of permissions, revoke privileges and remove users in a MySQL database.