**Creating user id DB**

CREATE USER 'abc'@'localhost' IDENTIFIED BY 'abc';



GRANT ALL PRIVILEGES ON \* . \* TO 'abc'@'localhost';

**SELECT**

**user**

**FROM**

mysql.user;

Under some circumstances, [CREATE USER](https://dev.mysql.com/doc/refman/5.7/en/create-user.html) may be recorded in server logs or on the client side in a history file such as ~/.mysql\_history, which means that cleartext passwords may be read by anyone having read access to that information.

mysql -u root -p

Enter password: \*\*\*\*\*\*\*\*\*\*\*

mysql> **use** mysql;

Database changed

mysql> **SELECT** **user** **FROM** **user**;

To get more information on the user table, you can preview its columns using the following command:

DESC user;

For example, to show users and other information such as host, account locking, and password expiration status, you use the following query:

**SELECT**

**user**,

host,

account\_locked,

password\_expired

**FROM**

mysql.**user**;

Show current user

To get the information on the  current user, you use the user() function as shown in the following statement:

**SELECT** **user**();

Or you use the current\_user() function:

**SELECT** **current\_user**();

Switch user:

SYSTEM mysql -u user -p

Check the grants assigned

SHOW GRANTS;

SHOW GRANTS FOR 'abc1'@'localhost';

## 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:

GRANT type\_of\_permission ON database\_name.table\_name TO 'username'@'localhost';

GRANT SELECT, INSERT ON \*.\* TO u1;

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';

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

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

DROP USER 'username'@'localhost';

**Creating Roles in Database:**

CREATE ROLE 'r1', 'r2';

GRANT SELECT ON db1.\* TO 'r1';

GRANT INSERT, UPDATE, DELETE ON db1.\* TO 'r2';

CREATE USER 'u1'@'localhost' IDENTIFIED BY 'u1pass';

GRANT 'r1', 'r2' TO 'u1'@'localhost';

SHOW GRANTS FOR 'u1'@'localhost';

Adding a USING clause causes the statement to also display the privileges associated with each role named in the clause:

SHOW GRANTS FOR 'u1'@'localhost' USING 'r1';

SHOW GRANTS FOR 'u1'@'localhost' USING 'r2';

SHOW GRANTS FOR 'u1'@'localhost' USING 'r1', 'r2';

**Note**

A privilege granted to an account is always in effect, but a role is not. The active roles for an account can differ across and within sessions, depending on the value of the [activate\_all\_roles\_on\_login](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html" \l "sysvar_activate_all_roles_on_login) system variable, the account default roles, and whether [SET ROLE](https://dev.mysql.com/doc/refman/8.0/en/set-role.html) has been executed within a session.

#### ALTER USER Statement

ALTER USER 'bharati'@'localhost'

IDENTIFIED BY 'abcd1'

ALTER USER 'abc1'@'localhost' PASSWORD EXPIRE INTERVAL 180 DAY;

ALTER USER 'abc'@'localhost' PASSWORD REUSE INTERVAL 360 DAY;

ALTER USER 'abc'@'localhost'

FAILED\_LOGIN\_ATTEMPTS 4 PASSWORD\_LOCK\_TIME 2;

ALTER USER 'abc'@'localhost' ACCOUNT LOCK;

ALTER USER 'abc'@'localhost' ACCOUNT UNLOCK;

RENAME USER 'abc'@'localhost' TO 'xyz';

* PASSWORD EXPIRE

Immediately marks the password expired for all accounts named by the statement.

ALTER USER 'abc'@'localhost' PASSWORD EXPIRE;

* PASSWORD EXPIRE DEFAULT

Sets all accounts named by the statement so that the global expiration policy applies, as specified by the [default\_password\_lifetime](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html" \l "sysvar_default_password_lifetime) system variable.

ALTER USER 'abc'@'localhost' PASSWORD EXPIRE DEFAULT;

* PASSWORD EXPIRE NEVER

This expiration option overrides the global policy for all accounts named by the statement. For each, it disables password expiration so that the password never expires.

ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE NEVER;

REVOKE role FROM user;

DROP ROLE 'r1', 'r2';