1. **Quick Overview of MySQL User Privileges:**
2. **Privilege Types**

* Column level: Control access to specific columns
* Table level: Manage permissions on entire tables
* Database level: Set permissions across all tables in a database
* Global level: Control server-wide permissions

1. **Key Operations:**

* GRANT - Assign permissions
* REVOKE - Remove permissions
* SHOW GRANTS - View current permissions
* FLUSH PRIVILEGES - Reload privilege tables into memory

1. **Common Privileges:**

* SELECT - Read data
* INSERT - Add records
* UPDATE - Modify records
* DELETE - Remove records
* CREATE - Create tables/databases
* DROP - Delete tables/databases
* ALTER - Modify table structure
* EXECUTE - Run stored procedures

1. **Syntax Pattern:**  
   GRANT privilege\_type ON database\_name.table\_name TO 'username'@'hostname' [WITH GRANT OPTION];
2. **User Management**

a) **Creating a Standard User**

**CREATE USER 'username'@'host' IDENTIFIED BY 'password';**

🡪 Creates a new MySQL user with the specified username and password. 'host' specifies the allowed connection source (e.g., 'localhost', '%' for any host, or a specific IP).

1. **Managing User Accounts**

* **Changing User Password :**

**ALTER USER 'username'@'host' IDENTIFIED BY 'new\_password';**

🡪 Updates the user’s password.

* **Locking and Unlocking User Accounts:** You can lock a user account to prevent it from logging in and unlock it when you need to restore access.

**ALTER USER 'username'@'host' ACCOUNT LOCK;**

**ALTER USER 'username'@'host' ACCOUNT UNLOCK;**

🡪 Locks or unlocks the user account, preventing or allowing login.

* **Deleting a User:** Removes a user account and all associated privileges from the MySQL server.

**DROP USER 'username'@'host';**

🡪 Removes the user from the MySQL database.

**3. Permission Management**

a) **Assigning Permissions**

* **Granting Basic Privileges:** Assigns a specific privilege (e.g., SELECT, INSERT, UPDATE) on a particular database and table to a user.

**GRANT privilege ON database.table TO 'username'@'host';**

🡪 Grants a specific privilege (e.g., SELECT, INSERT) on a database table.

* **Granting Multiple Privileges:** Allows you to grant multiple privileges at once on all tables within a database.

**GRANT SELECT, INSERT, UPDATE, DELETE ON db.\* TO 'user'@'host';**

🡪 Grants multiple permissions on all tables in the specified database.

* **Granting Column-Level Permissions:** You can restrict user access to certain columns in a table.

**GRANT SELECT (column1, column2) ON db.table TO 'user'@'host';**

🡪 Grants permission to access specific columns in a table.

b) **Viewing and Revoking Permissions**

* **Viewing Permissions:** Displays all the privileges assigned to a particular user.

**SHOW GRANTS FOR 'username'@'host';**

🡪 Displays privileges assigned to a specific user.

* **Revoking Permissions**

**REVOKE privilege ON database.table FROM 'username'@'host';**

🡪 Removes a specific privilege from a user.

1. **System Maintenance**

a) **Refreshing Privileges:** If you manually edit MySQL system tables (e.g., mysql.user) instead of using the GRANT or REVOKE commands, you should reload the privilege tables so that changes take effect immediately.

**FLUSH PRIVILEGES;**

🡪 Reloads the grant tables to apply changes immediately.

b) **Monitoring Connections:** Check the active queries and sessions to troubleshoot performance issues and detect locked or long-running processes. This also helps to identify unauthorized access or excessive connections.

**SHOW PROCESSLIST;**

🡪 Displays active MySQL connections and queries.

**5. Auditing Commands**

a) **Listing Users:**

**SELECT user, host FROM mysql.user;**

🡪 Lists all MySQL users and their allowed connection sources.

b) **Exporting Privileges:**

**SELECT \* FROM mysql.user WHERE user NOT LIKE 'mysql.%';**

**SELECT \* FROM mysql.db WHERE user NOT LIKE 'mysql.%';**

🡪 Retrieves non-system user accounts and their privileges.

**6. Emergency Procedures**

a) **Quick Locking of User Account:** Immediately disable a user account and remove all its privileges to prevent any further access—crucial in a security incident.

**ALTER USER 'username'@'host' ACCOUNT LOCK;**

**REVOKE ALL PRIVILEGES ON \*.\* FROM 'username'@'host';**

🡪 Immediately disables the user’s account and revokes all privileges.

b) **Resetting Permissions:** Completely remove existing privileges from a user and then reassign minimal or specific access as needed.

**REVOKE ALL PRIVILEGES, GRANT OPTION FROM ‘username’@’host’;**

**GRANT SELECT ON db.\* TO ‘username’@’host’;**

🡪 Removes all privileges and reassigns minimal access.