### ****Manually Rename the Database****

#### ****Backup the Existing Database****

1. Create users <https://www.geeksforgeeks.org/mysql-create-user-statement/?ref=lbp>

In MySQL, you **cannot** rename a database directly using an SQL command like RENAME DATABASE. Instead, you need to follow these steps:

https://www.atlassian.com/data/admin/how-to-rename-a-database-in-mysql

### ****Method 1: Manually Rename the Database****

**Create a New Database**

CREATE DATABASE class312;

**Copy All Tables from the Old Database** For each table in class310, run:

CREATE TABLE class312.table\_name LIKE class310.table\_name;

INSERT INTO class312.table\_name SELECT \* FROM class310.table\_name;

You can automate this using a script.

**Drop the Old Database (After Verifying Data)**

DROP DATABASE class310;

**⚠️ Warning:** This will permanently delete class310, so ensure you have a backup.

### ****Method 2: Using MySQL Dump****

If you have many tables, the best way is to use a **dump and restore** approach.

#### ****Step 1: Backup the Existing Database****

Run this in the command line:

mysqldump -u root -p class310 > class310\_backup.sql

#### ****Step 2: Create the New Database****

CREATE DATABASE class312;

#### ****Step 3: Restore Data to the New Database****

mysql -u root -p class312 < class310\_backup.sql

#### ****Step 4: Drop the Old Database (Optional)****

Once you verify everything, drop the old database:

DROP DATABASE class310;

There is no direct RENAME DATABASE command in MySQL, so one of these methods must be used. Let me know if you need help automating this! 🚀

### ****Backing Up a MySQL Database****

MySQL provides several ways to back up databases. The most common method is using the mysqldump command.

## ****1. Backup Using**** mysqldump ****(Recommended)****

### ****Basic Backup (Single Database)****

Run this in the command line:

mysqldump -u root -p database\_name > backup.sql

* database\_name → Replace with your actual database name.
* backup.sql → This is the output SQL file where the backup is stored.
* You'll be prompted to enter the MySQL password.

### ****2. Backup Multiple Databases****

mysqldump -u root -p --databases db1 db2 db3 > backup.sql

* Replace db1 db2 db3 with actual database names.

### ****3. Backup All Databases****

mysqldump -u root -p --all-databases > all\_databases\_backup.sql

This backs up all databases into a single file.

### ****4. Backup with Data Only (No Structure)****

If you only need the data (no table structure):

mysqldump -u root -p --no-create-info database\_name > data\_backup.sql

### ****5. Backup with Structure Only (No Data)****

If you only need table structures:

mysqldump -u root -p --no-data database\_name > structure\_backup.sql

### ****6. Backup with Compression (Smaller File)****

To reduce file size, use gzip:

mysqldump -u root -p database\_name | gzip > backup.sql.gz

To restore a compressed backup:

gunzip < backup.sql.gz | mysql -u root -p database\_name

## ****Restoring a Backup****

To restore a MySQL backup:

mysql -u root -p database\_name < backup.sql

* Ensure database\_name exists before running this command:

CREATE DATABASE database\_name;

## ****Automating Backup (Cron Job)****

To schedule automatic backups (Linux):

crontab -e

Add this line to run a backup daily at midnight:

0 0 \* \* \* mysqldump -u root -p'yourpassword' database\_name > /path/to/backup.sql

**⚠️ Note:** Avoid storing passwords in scripts; use a .my.cnf file for security.

Let me know if you need a step-by-step guide for your setup! 🚀

### ****Creating a New User in MySQL****

You can define a new user in MySQL using the CREATE USER statement and grant appropriate privileges.

### ****1. Create a New User****

Syntax:

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

Example:

CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'SecurePass123';

* 'newuser' → The username you want to create.
* 'localhost' → The host from which the user can connect (use '%' for any host).
* 'SecurePass123' → The password for the user.

### ****2. Grant Privileges to the User****

To give full privileges on a specific database:

GRANT ALL PRIVILEGES ON database\_name.\* TO 'newuser'@'localhost';

To grant all privileges on **all** databases:

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

If you only want to grant SELECT, INSERT, and UPDATE:

GRANT SELECT, INSERT, UPDATE ON database\_name.\* TO 'newuser'@'localhost';

After granting privileges, apply changes:

FLUSH PRIVILEGES;

### ****3. Verify the User****

To check existing users:

SELECT user, host FROM mysql.user;

### ****4. Remove a User****

To delete the user:

DROP USER 'newuser'@'localhost';

### ****5. Change User Password****

If you need to update the password:

ALTER USER 'newuser'@'localhost' IDENTIFIED BY 'NewSecurePass456';

### ****6. Test the New User****

Try logging in with:

mysql -u newuser -p -h localhost

Enter the password when prompted.

Let me know if you need further help! 🚀