

SQL - Rename Database

There can be several reasons to rename a database name. One of the reasons could be to avoid naming conflicts or to separate different types of data into different databases. Another reason can be to arrange them in an organized way which makes them more descriptive and easier to manage.

Rename Database using SQL

SQL provides a simple **RENAME DATABASE...TO** statement to rename an existing database. If you want to rename a database, make sure there is no active transaction in progress otherwise the complete operation might halt once you rename the database.

RENAME DATABASE...TO (Obsolete)

The **SQL RANME DATABASE...TO** statement is used to rename an existing user-created database.

Syntax

Following is the syntax of the RENAME DATABASE...TO statement -

```
RENAME DATABASE OldDatabaseName TO NewDatabaseName;
```

Example

```
SHOW DATABASES;
```

The output will be displayed as -

Database
performance_schema
information_schema
mysql
testDB

Now, issue the following command to rename the database **testDB** to **tutorialsDB**:

```
RENAME DATABASE testDB TO tutorialsDB;
```

There used to be a simple RENAME DATABASE command in older versions of MySQL which was intended to rename database but RENAME DATABASE command has been removed from all newer versions to avoid security risks. Next section will show you a work around to rename your MySQL database.

Rename Database using Dump and Reimport

If you are willing to rename a database name in MySQL, then simple way is to dump the complete database in an SQL file and then re-import it into a new database. This is three step process which we will follow in this tutorial:

Step 1 - Dump Old Database

Consider you want to rename **testDB** database to **tutorialsDB**. So first we will dump it in a simple SQL file using MySQL **mysqldump** command. This operation will be performed from command line and will require a database user name and password, preferably admin privilege.



We are using the `-p` flag immediately followed by our password to connect to the database with no space between. The `-R` is required to tell **mysqldump** to copy stored procedures and functions along with the normal data from the database.

Step 2 – Create New Database

Next step is to create our new database using **mysqladmin** prompt command as follows:

```
$ mysqladmin -u username -p"password" create tutorialsDB;
```

Step 3 – Import Old Database

The final step is to import old database into new database as following:

```
$ mysql -u username -p"password" tutorialsDB < testDB.sql;
```

Step 4 – Verification (Optional)

Now you can verify the changes by listing down all the available databases:

```
SHOW DATABASES;
```

The output will be displayed as –

Database
performance_schema
information_schema
mysql
testDB

Step 5 - Verification (Optional)

Once you are satisfied with your changes, you can delete your old database as follows:

```
DROP DATABASE testDB;
```

The ALTER DATABASE...MODIFY Statement

The **ALTER DATABASE...MODIFY** statement in SQL is used to rename the name of an existing database name in SQL Database Server. Please note that this command does not work in MySQL database.

Syntax

Following is the syntax of the ALTER DATABASE...MODIFY command -

```
ALTER DATABASE OldDatabaseName MODIFY NAME = NewDatabaseName;
```

Example

Following is the SQL command in SQL Server to rename the database testDB to tutorialsDB:

```
ALTER DATABASE testDB MODIFY NAME = tutorialsDB ;
```

Kickstart Your Career

Get certified by completing the course

[Get Started](#)

