

# LU09.A02 - SQL- DCL: CREATE USER

## Task 2.1

**Create the User:** Create a user named *restrictedUser* with the password *SafePassword123* using the *mysql\_native\_password* plugin.

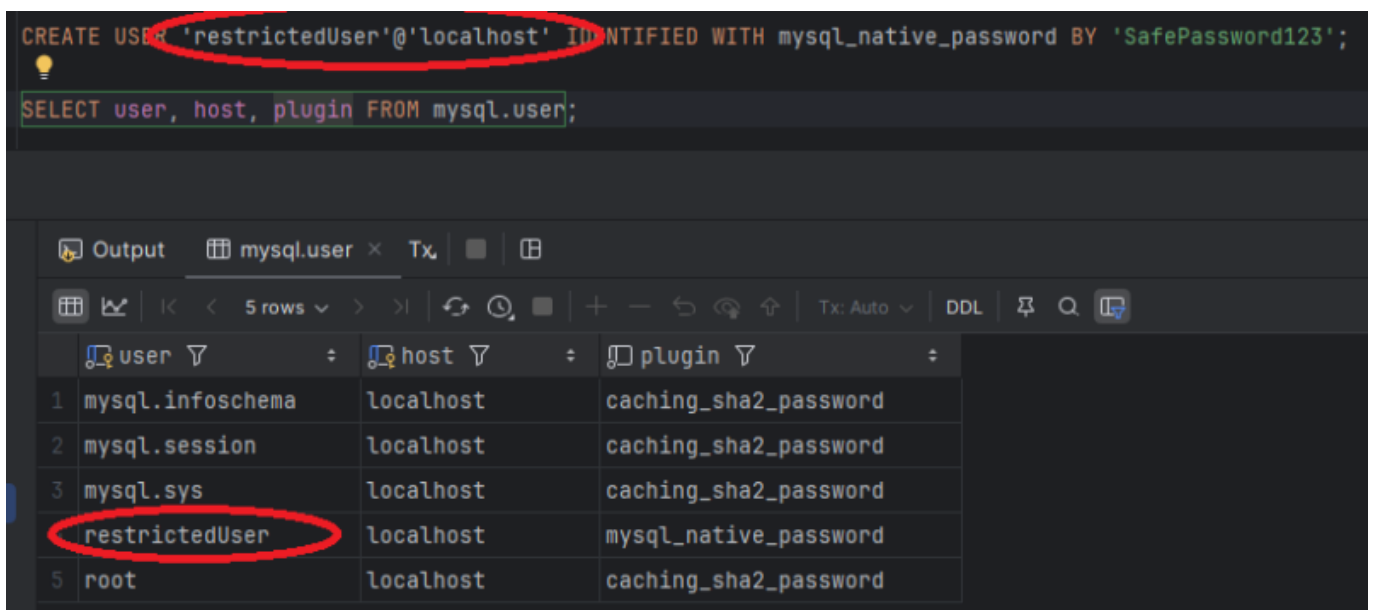
```
CREATE USER 'restrictedUser'@'localhost' IDENTIFIED WITH mysql_native_password BY 'SafePassword123';
```

## Task 2.2

**Overview of the current privileges:** Display all users.

```
SELECT user, host, plugin FROM mysql.user;
```

The result set should look like:



The screenshot shows a MySQL command-line interface. The first command executed is `CREATE USER 'restrictedUser'@'localhost' IDENTIFIED WITH mysql_native_password BY 'SafePassword123';`, where the user and host are circled in red. The second command is `SELECT user, host, plugin FROM mysql.user;`. The output is a table with 5 rows. The third row, representing the newly created user, is circled in red.

	user	host	plugin
1	mysql.infoschema	localhost	caching_sha2_password
2	mysql.session	localhost	caching_sha2_password
3	mysql.sys	localhost	caching_sha2_password
4	restrictedUser	localhost	mysql_native_password
5	root	localhost	caching_sha2_password

## Task 2.3

**Grant Privileges Without Table Management:** Grant the user *restrictedUser* the ability to read and write data but not to create, alter, or drop tables. Use the following commands to give only the required privileges.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON myDatabase.* TO 'restrictedUser'@'localhost';
```

## Task 2.4

**Revoke Privileges:** To be certain that nothing unintended can happen revoke the CREATE, ALTER, and DROP privileges explicitly.

```
REVOKE CREATE, ALTER, DROP ON myDatabase.* FROM 'restrictedUser'@'localhost';
```

## Task 2.5

**View the User's Privileges:** Check the privileges to ensure that the user cannot manage tables.

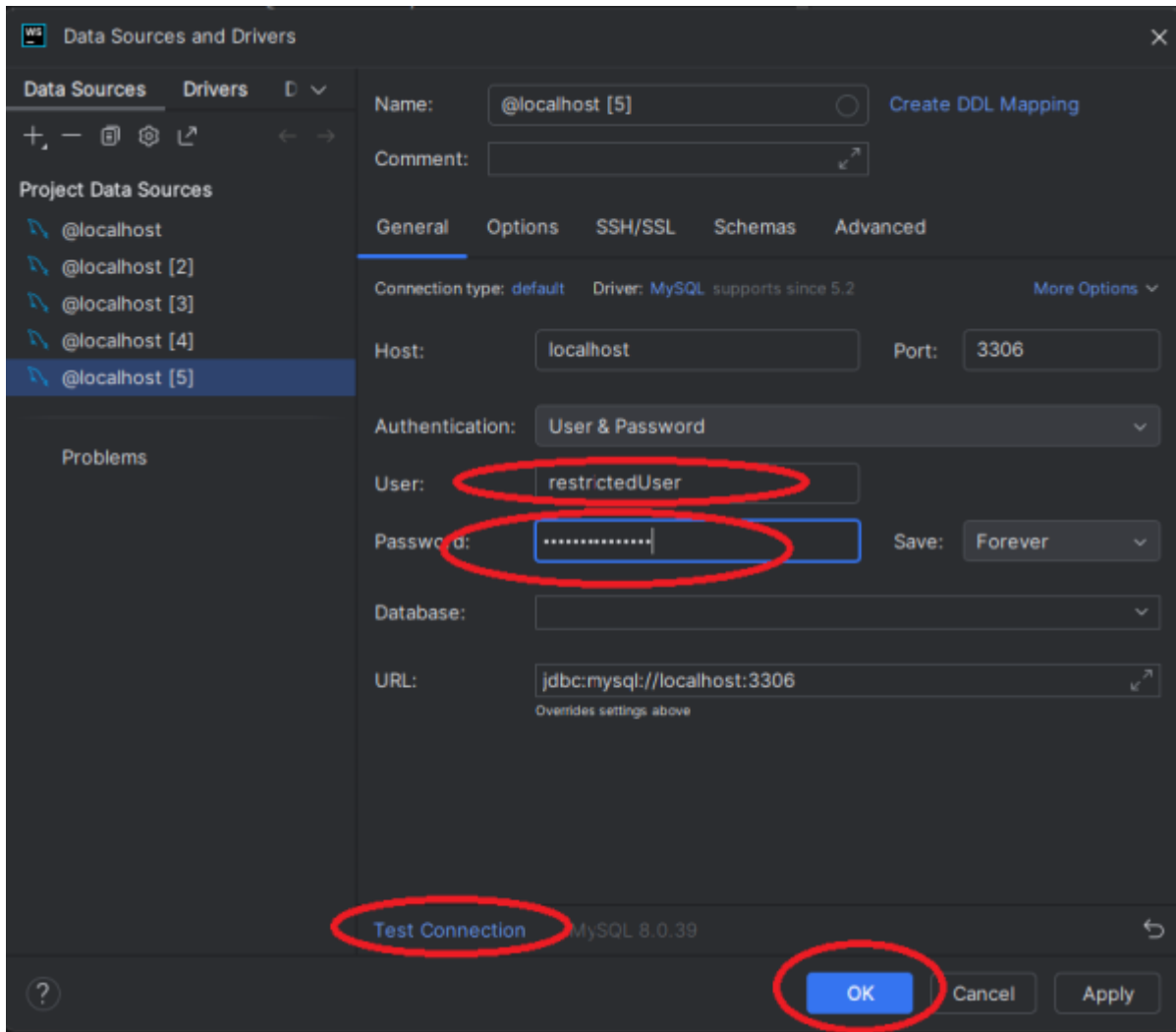
```
SHOW GRANTS FOR 'restrictedUser'@'localhost';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON myDatabase.* TO 'restrictedUser'@'localhost';
REVOKE CREATE, ALTER, DROP ON myDatabase.* FROM 'restrictedUser'@'localhost';
SHOW GRANTS FOR 'restrictedUser'@'localhost';
```

	Grants for restrictedUser@localhost
1	GRANT USAGE ON *.* TO 'restrictedUser'@'localhost'
2	GRANT SELECT, INSERT, UPDATE, DELETE ON `mydatabase`.* TO 'restrictedUser'@'localhost'

## Task 2.6

**Test the User's Access:** Establish a new console connection to the database by using *restrictedUser* + *password*.



## Task 2.7

Finally, try to perform a CREATE TABLE or DROP TABLE operation. The attempt should result in a permission error also displayed in the image below.

The screenshot shows a database IDE with a SQL console and a table view. The console contains the following SQL commands:

```
1 USE myDatabase;  
2  
3 SELECT * FROM users;  
4  
5 DROP TABLE users;
```

The fifth line, `DROP TABLE users;`, is highlighted with a red arrow. A red arrow also points to the error message in the console:

```
[42000][1142] DROP command denied to user 'restrictedUser'@'localhost' for table 'users'
```

The Database Explorer on the right shows the server structure, with a red arrow pointing to the `users` table under the `mydatabase` schema.

The Output window shows the result of the `SELECT * FROM users;` query, displaying 20 rows of user data:

user_id	username	email
1	john_doe	john.doe@example.com
2	jane_smith	jane.smith@example.com
3	michael_brown	michael.brown@example.com
4	sarah_johnson	sarah.johnson@example.com
5	chris_williams	chris.williams@example.com
6	anna_lee	anna.lee@example.com
7	david_kim	david.kim@example.com
8	laura_clark	laura.clark@example.com
9	jake_lewis	jake.lewis@example.com
10	emily_martinez	emily.martinez@example.com
11	john_doe	john.doe@example.com
12	jane_smith	jane.smith@example.com
13	michael_brown	michael.brown@example.com
14	sarah_johnson	sarah.johnson@example.com
15	chris_williams	chris.williams@example.com
16	anna_lee	anna.lee@example.com
17	david_kim	david.kim@example.com
18	laura_clark	laura.clark@example.com
19	jake_lewis	jake.lewis@example.com
20	emily_martinez	emily.martinez@example.com

## Task 2.8

**Delete the User (optional):** After testing, you can delete the user if they are no longer needed.

```
DROP USER 'restrictedUser'@'localhost';  
SELECT user, host, plugin FROM mysql.user;
```

	user	host	plugin
1	mysql.infoschema	localhost	caching_sha2_password
2	mysql.session	localhost	caching_sha2_password
3	mysql.sys	localhost	caching_sha2_password
4	root	localhost	caching_sha2_password

## Vocabulary

English	German
..	..



Volkan Demir

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/m290/learningunits/lu06/loesungen/l02?rev=1730201596>

Last update: **2024/10/29 12:33**

