

LU09.A02 - SQL- DCL: CREATE USER

It is high time to mess a little around with our new SQL powers, don't you think? So, let's try it directly on our Webstorm.

As the database administrator, we want to create a new user and only grant this user the necessary rights to operate the web application, which includes DML operations such as INSERT, UPDATE, DELETE, but not DDL operations such as CREATE or DROP of tables. After all, we don't want the web application to take control over our database, are we?

Requirements

- Work type: individual
- Timeframe: 10 Minutes
- Means of aid:
 - only teaching materials, no websearch, no use of ai.
 - Webstorm with connection to the MySQL-DB
- Expected result: Semantically and syntactically correct SQL statements according to the requirements of the case studies.

Case studies / Assignments

As a database administrator we are assigned to create a AppUser, which has for security reasons only the right for DML operations, but must not be allowed for DDL operation. We don't want a hacker to delete our entire webshop, do we?

To get the job done, follow the instructions below:

1. Create as the sysdba (systemadministrator of the database) a new user
2. Grant this role only the necessary rights
3. Create as a sysdba a test table and fill it with some testdata
4. Establish a new connection within the webstorm by using the credentials of this new user
5. Try out the DML operations, which should be possible (insert, update, delete)
6. Try out DDL operations, which should result in errors due to missing permissions for that particular user
7. Drop the newly created user finally

Task 2.1

Create the User: Create a user named *restrictedUser* with the password *SafePassword123* using the *mysql_native_password* plugin.

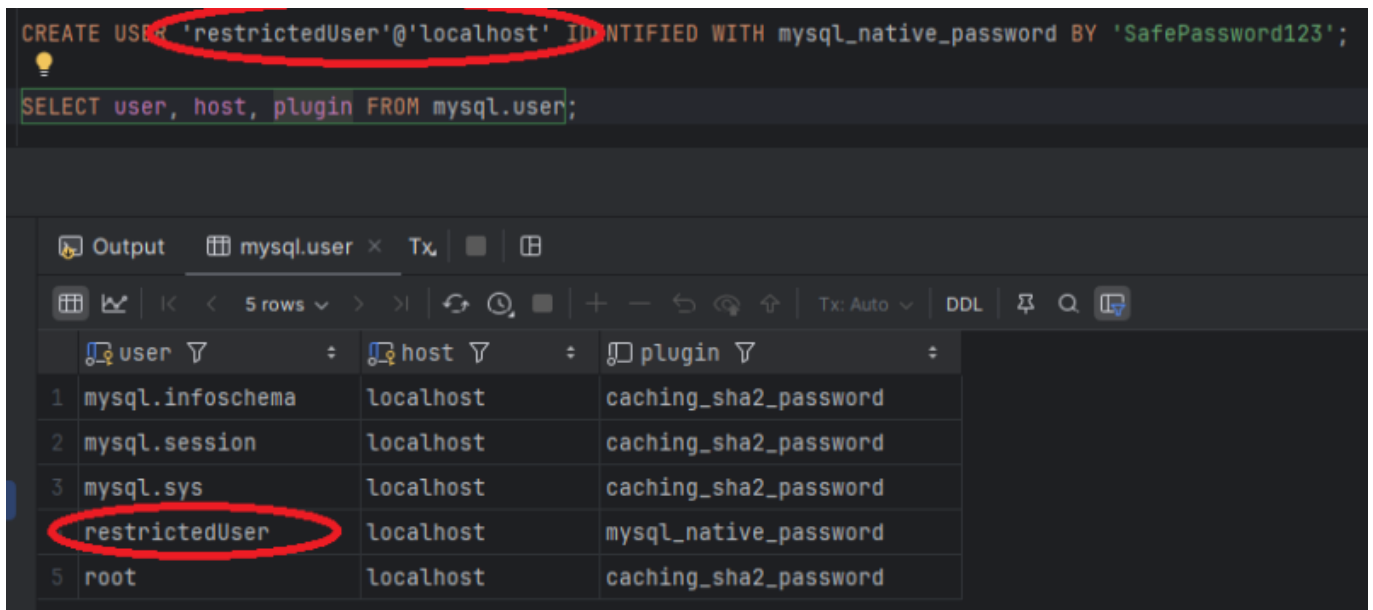
```
CREATE USER 'restrictedUser'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'SafePassword123';
```

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 terminal window. The command `CREATE USER 'restrictedUser'@'localhost' IDENTIFIED WITH mysql_native_password BY 'SafePassword123';` is highlighted with a red circle. Below it, the command `SELECT user, host, plugin FROM mysql.user;` is entered. The terminal output shows a table with 5 rows. The row for `restrictedUser` is highlighted with a red circle.

	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

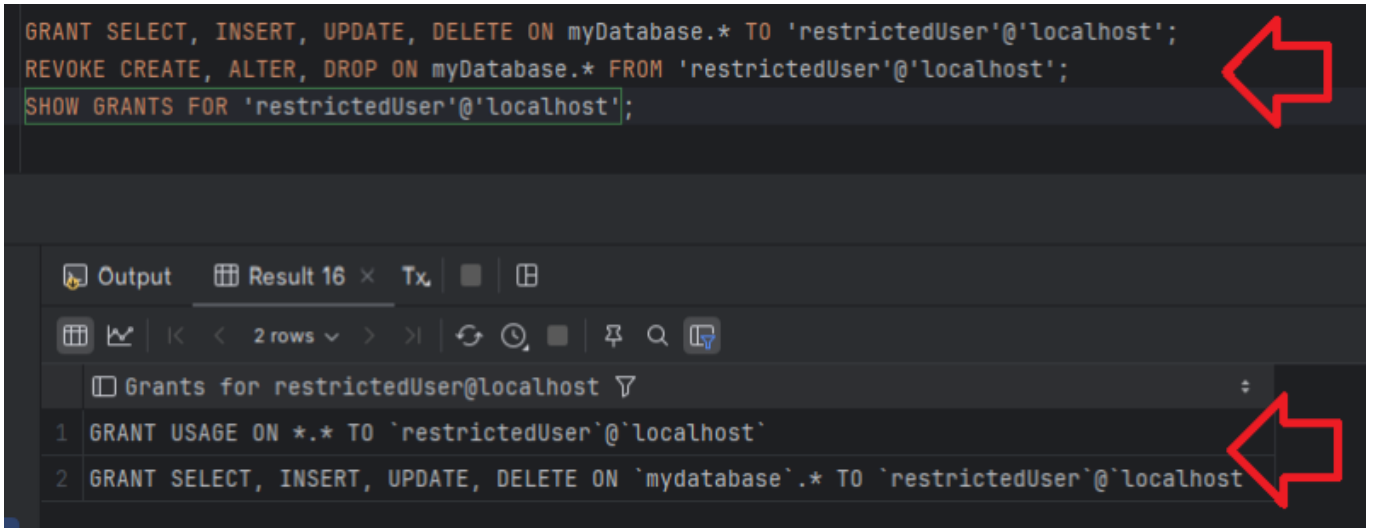
Revoke Privileges: To be certain that nothing unintended can happen revoke the CREATE, ALTER, and DROP privileges explicitly.

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

Task 2.5

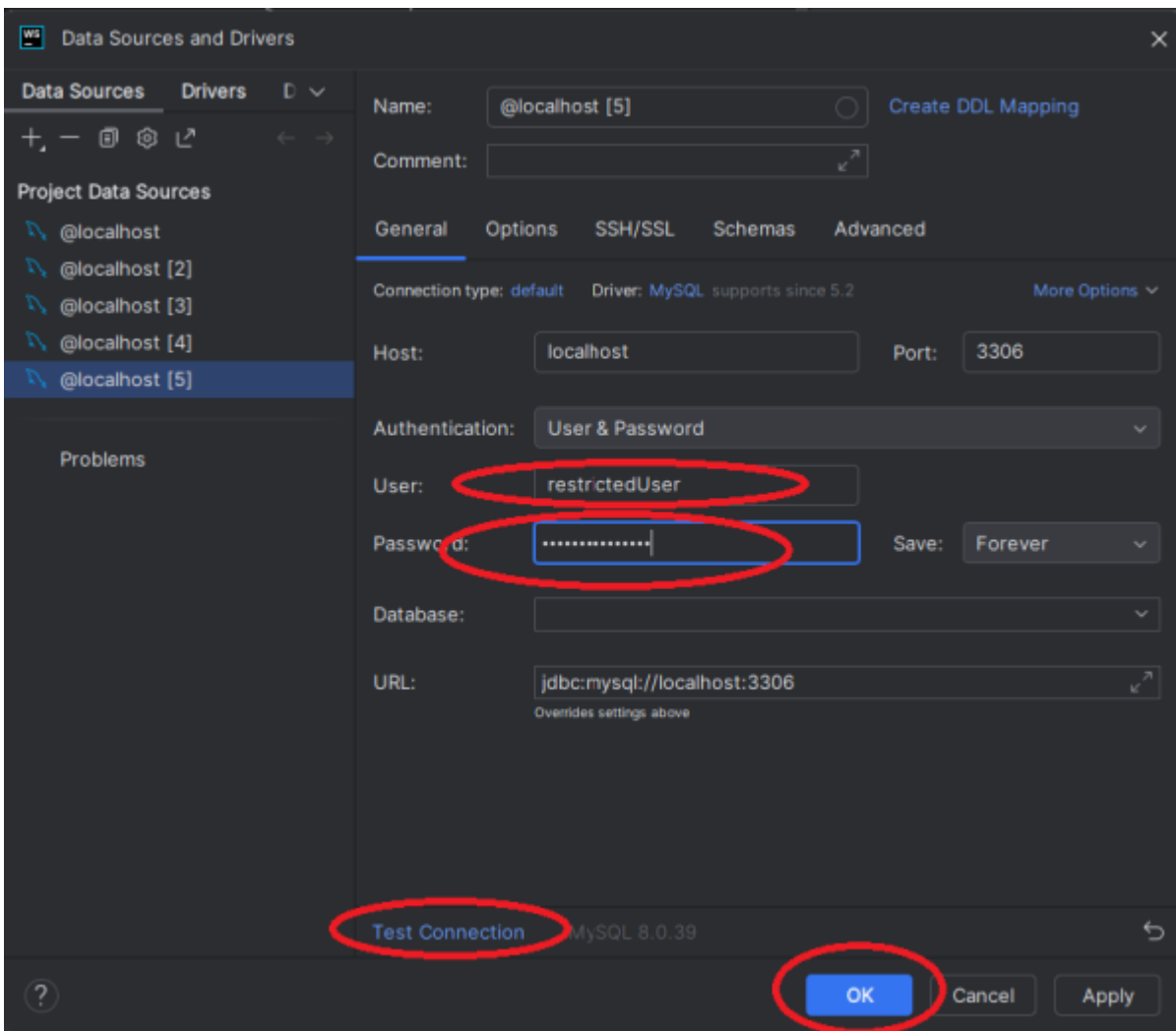
View the User's Privileges: Check the privileges to ensure that the user cannot manage tables.

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



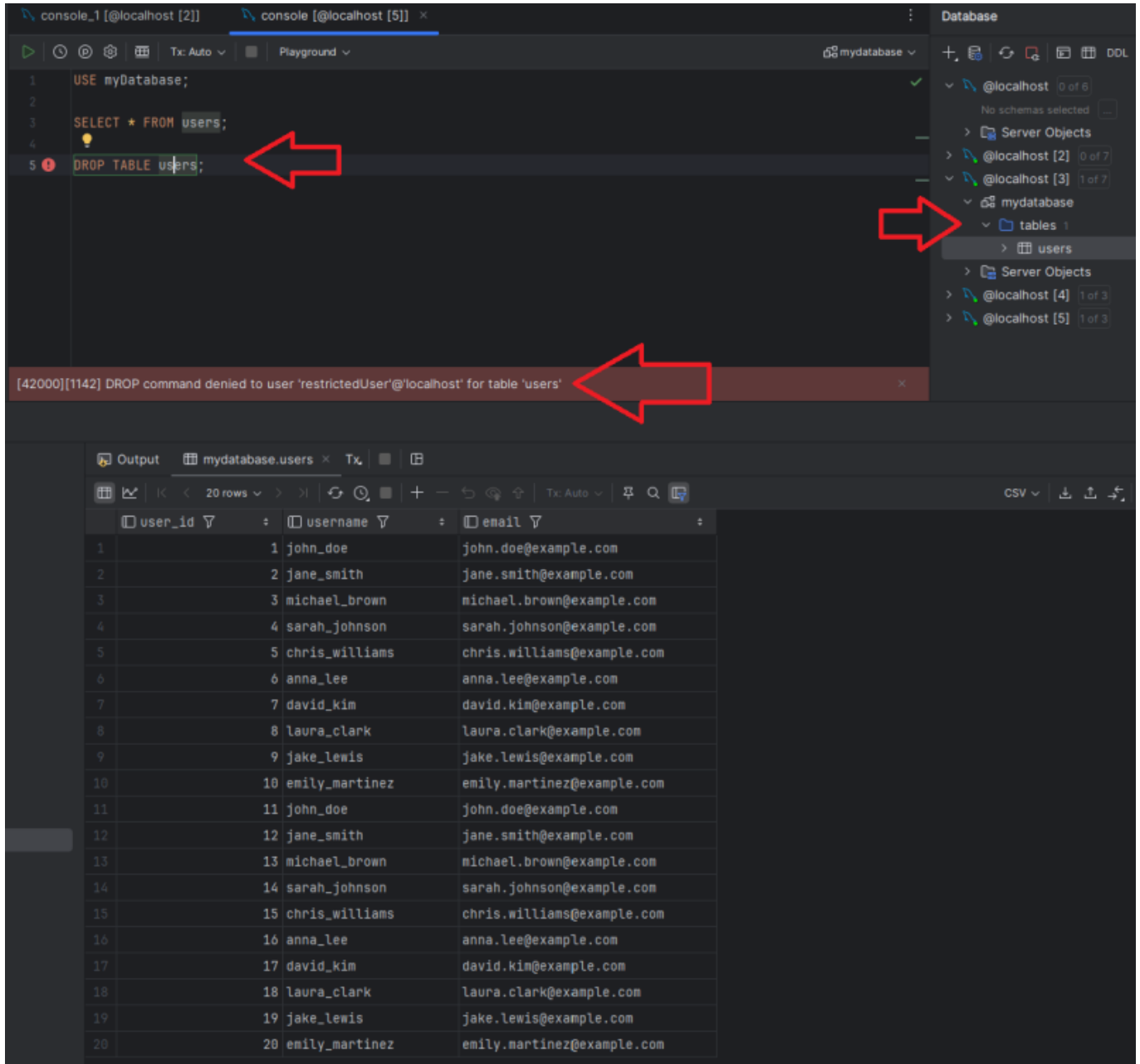
Task 2.6

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 als displayed in the image below.



The screenshot shows a database console interface with the following SQL commands:

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

A red arrow points to the error message at the bottom of the console:

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

Another red arrow points to the 'users' table in the database explorer on the right side of the interface.

Below the console, the output of the SELECT query is displayed in a table:

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=1730201536>

Last update: **2024/10/29 12:32**

