LU09a - USER and PRIVILEGE Management

Learning Objectives

1. Explain why different users are needed to access a database.
2. Create a mysql user on the command line and with a sql script
3. Grant required privileges to the database user newly created
4. Revoke privileges from user
5. Deleting obsolete user

Introduction

In a warehouse such as Globus, Manor or Lafayette there are different user categories with different authorizations and privileges. According to the required responsibilties the general manager e.g. has full privileges, while the apprentice has much fewer. And as we know, a database is basicly a warehouse, not for goods, but for data.

MySQL user management and privileges are essential for controlling access to the database system, ensuring security, and managing user roles. MySQL allows database administrators (DBAs) to create, manage, and assign permissions to users, limiting what actions users can perform on the database. This control helps protect sensitive data, prevent unauthorized changes, and maintain the overall integrity of the database.

Key Concepts

1. **User Accounts**: MySQL user accounts are created to define who can log in to the MySQL server and what they can access. Each user is identified by a username and a host, which specifies the IP address or domain from which the user can connect.
2. **Privileges**: MySQL privileges determine what actions a user can perform. These actions include SELECT (read), INSERT, UPDATE, DELETE, and more. Privileges can be granted at different levels, such as the global level, database level, table level, or even column level.
3. **Roles**: MySQL supports roles, which are collections of privileges that can be granted to users as a group. This simplifies managing permissions, especially in large organizations.
4. **Authentication**: MySQL supports various authentication methods, including password-based and plugin-based authentication, to verify user identities when they attempt to connect.

Common MySQL User Management Commands

1. Creating a New User

To create a new user in MySQL, you use the CREATE USER statement:

```
CREATE USER 'username'@'localhost' IDENTIFIED WITH 'caching_sha2_password'
```

```
BY 'password';
```

This creates a user username that can only connect from localhost (the MySQL server machine).

Note

MySQL supports several types of password authentication mechanisms. These include:

1. **mysql_native_password**: This is the traditional password hashing method in MySQL, using the SHA1 hashing algorithm. It's one of the oldest and widely used methods for authentication.
2. **caching_sha2_password**: This is the default authentication plugin starting from MySQL 8.0. It uses SHA-256 for hashing passwords and offers better security than mysql_native_password.
3. **sha256_password**: This plugin provides SHA-256 based password hashing and can be used in conjunction with SSL for encrypted communication.
4. **auth_socket**: This plugin allows users to authenticate based on the operating system's user credentials, which can be useful for local access without passwords (e.g., root user in some setups).
5. **auth_pam**: This plugin enables MySQL to use external authentication mechanisms like PAM (Pluggable Authentication Modules), which allows for integration with OS-level authentication systems or LDAP.
6. **authentication_ldap_sasl**: This plugin provides authentication using LDAP servers via the SASL (Simple Authentication and Security Layer) protocol.
7. **authentication_ldap_simple**: A simpler LDAP authentication plugin that doesn't use SASL but supports basic LDAP-based authentication.

2. Granting Privileges

To assign privileges to a user, the GRANT statement is used. For example, to grant a user the ability to SELECT, INSERT, and UPDATE on a specific database:

```
GRANT SELECT, INSERT, UPDATE ON database_name.* TO 'username'@'localhost';
```

The * symbol specifies that the privileges apply to all tables within database_name.

3. Viewing User Privileges

You can check a user's privileges with the SHOW GRANTS command:

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

This displays all the privileges granted to the user username.

4. Revoking Privileges

To remove privileges from a user, the REVOKE statement is used. For example, to revoke the INSERT privilege:

```
REVOKE INSERT ON database_name.* FROM 'username'@'localhost';
```

5. Dropping a User

If a user is no longer needed, you can remove the account using the DROP USER statement:

```
DROP USER 'username'@'localhost';
```

This removes both the user and their associated privileges.

6. Activating Change of Privileges After Altering Them

To activate changes to privileges in MySQL after modifying them, you can use the following command:

```
FLUSH PRIVILEGES;
```

This command forces MySQL to reload the privilege tables, applying any recent changes made to user privileges. Normally, after using GRANT, REVOKE, or ALTER USER, MySQL applies changes automatically, but if you've made manual adjustments directly in the mysql database tables, running FLUSH PRIVILEGES ensures that the changes take effect immediately.

Please note that the change will only take effect AFTER a new login. This means that you must open a new console window with your new login data in order to recognize the change in permissions.

Video-Tutorials

How to Create a New User in MySQL	how_to_create_a_new_user_in_mysql.mp4
Create User, Grant ReadWrite Privileges and Test Privileges Granted in MySQL	create_user_grant_readwrite_privileges_and_test_privileges_granted_in_mysql.mp4

Vocabulary

English	Deutsch
obsolete	veraltet
commodity	Handelsgut, Ware
to revoke	widerrufen, aufheben

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m290/learningunits/lu06/theorie/01

Last update: **2024/11/16 09:00**