Users and Privileges

Soma Soleiman Zadeh

Database Systems II (IT 216)

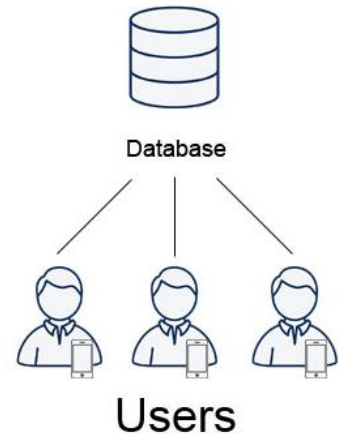Spring 2024 - 2025

Week 13

May 6, 2025

Outline

○ **USER** in MySQL

○ Syntax of Creating a **USER** in MySQL

○ **Privileges**

- **Granting** Privileges to Users

- **Revoking** Privileges from Users

○ Deleting a **USER**

What is a Database User?

◦ A Database User is defined as a person who interacts with data daily, updating, reading, and modifying the given data.

◦ The access and permissions within a database depends on their function and tasks.

Users in MySQL

◦ After installing MySQL, the first user created is the **root** user, which is the MySQL administrator.

◦ The **root** user will have permission to do everything on the MySQL databases.

◦ Sharing the **root** user between several people is dangerous. Hackers often try to log in as the **root** user, steal the information, or even destroy the whole MySQL server alongside its data.

◦ **Solution** → System administrators can create users with specific permissions on some databases.

Syntax of Creating a USER in MySQL

◦ You can create a new user with a **CREATE USER** statement, with the following syntax:

CREATE USER 'user_name'@'host_name' IDENTIFIED BY 'password';

◦ To create a new user in MySQL, specify the **username**, the **hostname** the user can use to access the database management system, and a secure **password**.

Syntax of Creating a USER in MySQL

◦ If you specify *localhost* as the host_name, the created user will access the MySQL instance from the local machine (localhost).

CREATE USER 'ali'@'localhost' IDENTIFIED BY 'abc123';

◦ If you want the created user, be able access the MySQL instance remotely from another machine, use a % character in the host_name definition.

CREATE USER 'ali'@'%' IDENTIFIED BY 'abc123';

CREATE USER 'ali'@'10.0.%' IDENTIFIED BY 'abc123';

How to See all Users in a MySQL Instance?

◦ The **mysql.user** table contains information about users that have permission to access the MySQL server, and their privileges.

◦ To see all **users** and their **hosts** in your MySQL instance:

SELECT user, host **FROM** **mysql.user**;

◦ How to see the current user:

SELECT **current_user()**;

User Privileges

◦ When different users can access database objects, the administrator can control how each user can deal with database objects by defining each user's privileges.

◦ A **Privilege** is <u>a permission or right granted to a user</u>, allowing them to perform specified actions (such as DELETE, UPDATE, etc.) on database objects.

Some User Privileges

Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
REFERENCES	Ability to create a constraint that refers to the table.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
INDEX	Ability to create an index on the table with the create index statement.
ALL	All privileges on table.

GRANT Privileges to a USER

◦ After creating a user, grant some privileges to the user by using:

- ◦ **GRANT** statement
- ◦ Main functionality of **GRANT** :
    - ◦ Managing the privileges of users.

◦ Syntax for **GRANT** privileges to Users:

GRANT privilege **ON** database.table **TO 'user_name'**@**'host_name'**;

GRANT SELECT ON hospital.patient TO 'ali'@**'localhost'**;

Review a User's Current Privileges

◦ You can review a user's current permissions by using SHOW GRANTS command:

SHOW GRANTS FOR 'user_name'@'host_name';

◦ For example, by running the following statement, all privileges of a user named **'ali'@'localhost'** is listed.

SHOW GRANTS FOR 'ali'@'localhost';

REVOKE Privileges from a USER

◦ After granting privileges to a specific user, you may need to take back some of those privileges from the user.

◦ Syntax of **REVOKE** statement:

REVOKE privileges **ON database.table FROM 'user_name'@'host_name'**;

REVOKE Privileges from a USER

◦ **Example:**

    GRANT DELETE, UPDATE ON hospital.patient TO 'ali'@'localhost';

    REVOKE DELETE ON hospital.patient FROM 'ali'@'localhost';

◦ According to this example, first **DELETE** and **UPDATE** privileges on **patient** table are given to user named **'ali'.**

◦ Then from the current privileges that **'ali'** user has, we take back his **DELETE** privilege on **patient** table.

Delete a USER

◦ You can delete a user by using DROP USER statement:

    DROP USER 'user_name'@'host_name';

◦ **Example:**

    DROP USER 'ali'@'localhost';