Lesson 7.5: Users and Permissions

CSC430/530 – DATABASE MANAGEMENT SYSTEMS

User Management

- Creating a User
  - 'host' specifies the allowed connection source (e.g., 'localhost', '%' for any host, or a specific IP

```
CREATE USER 'username'@'host' IDENTIFIED BY 'password';
```

- Changing User Password

```
ALTER USER 'username'@'host' IDENTIFIED BY 'new_password';
```

- Locking and Unlocking User Accounts
  - You can lock a user account to prevent them from logging in and unlock it when you need to restore access.

```
ALTER USER 'username'@'host' ACCOUNT LOCK;
ALTER USER 'username'@'host' ACCOUNT UNLOCK;
```

- Deleting a User:
  - Removes a user account and all associated privileges from the MySQL server.

```
DROP USER 'username'@'host';
```

Grant Privileges

- Syntax

```
GRANT privilege_type (column_list) ON item TO 'username'@'host';
```

- privilege_type
  - SELECT – enables use of SELECT command
  - INSERT - enables use of INSERT command
  - UPDATE - enables use of UPDATE command
  - DELETE - enables use of DELETE command
  - CREATE - enable database and table creation
  - DROP - enable databases, tables, and views to be dropped
  - ALTER – enables use of ALTER TABLE
  - EXECUTE - enable the user to execute stored procedures
  - ALL – Can do everything!
- column_list
  - Specify individual columns separated by commas if column level control is desired
- item
  - `*.*` used to specify server wide permissions
  - `database.*` used to specify database level permissions on specified database
  - `database.table` used to specify table level permissions on specified table within database

Examples

```sql
CREATE USER 'employee'@'localhost' IDENTIFIED BY 'pass123';
GRANT SELECT(fname, lname), UPDATE(fname)
ON company.employee
TO 'employee'@'localhost';


CREATE USER 'manager'@'localhost' IDENTIFIED BY 'sunshine42';
GRANT SELECT, UPDATE, DELETE, EXECUTE
ON company.*
TO 'manager'@'localhost';


CREATE USER 'ceo'@'localhost' IDENTIFIED BY 'iamceo';
GRANT ALL
ON *.*
TO 'ceo'@'localhost';
```

Grants

- Viewing Permissions
  - Displays all the privileges assigned to a particular user.

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

- Revoking Permissions
  - Removes a specific privilege from a user.

```
REVOKE privilege_type ON database.table FROM 'username'@'host';
```

- Refreshing Privileges
  - If you manually edit MySQL system tables (e.g., mysql.user) instead of using the GRANT or REVOKE commands, you should reload the privilege tables so that changes take effect immediately.

```
FLUSH PRIVILEGES;
```

- Monitoring Connections
  - Check the active MySQL connections and queries to troubleshoot performance issues and detect locked or long-running processes.
  - This also helps to identify unauthorized access or excessive connections.

```
SHOW PROCESSLIST;
```

Auditing

- Listing Users
  - Lists all MySQL users and their allowed connection sources

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

- Exporting Privileges
  - Retrieves non-system user accounts and their privileges.

    ```sql
    SELECT * FROM mysql.user WHERE user NOT LIKE 'mysql.%';
    SELECT * FROM mysql.db WHERE user NOT LIKE 'mysql.%';
    ```

Emergency!

- Quick Locking of User Account
  - Immediately disable a user account and revokes all its privileges to prevent any further access—crucial in a security incident.

    ```
    ALTER USER 'username'@'host' ACCOUNT LOCK;
    REVOKE ALL PRIVILEGES ON *.* FROM 'username'@'host';
    ```

- Resetting Permissions
  - Completely revokes existing privileges from a user and then reassigns minimal or specific access as needed.

    ```
    REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'username'@'host';
    GRANT SELECT ON db.* TO 'username'@'host';
    ```