MYSQL ACCESS CONTROLS

Access Denied!

USERS

- For the entire course up until this point, we have been logging in to the database as user root.

- Generally, this is not desirable behavior for a number of reasons:

  - One user and one password limits database use to one person, as sharing passwords is bad karma.

  - Security - root user by default can do *anything*.

  - Accountability.

CREATING USERS

- To create a new user, use the CREATE USER statement:

  -
    ```
    CREATE USER account IDENTIFIED BY ‘password’
    ```

- Example:

  -
    ```
    CREATE USER ‘moviedba’@‘localhost’ IDENTIFIED
    BY ‘popcorn’;
    ```

- This creates a new account *with no privileges* called “moviedba” with a password of “popcorn”, allowed to login from “localhost”.

PRIVILEGES

- The various actions that can be performed in MySQL are categorized into privileges, and the ability to perform an action or not is controlled by the privileges that have been granted to a user. Examples of some privileges:
  - `SELECT`: Issue SELECT statements
  - `INSERT`: Insert new data
  - `DROP`: Dropping structures within the database
  - `SHUTDOWN`: Allowed to initiate a shutdown of the server

PRIVILEGES

- The privileges granted can be limited in various ways:
  - Global: All databases and tables - think superuser
  - Database: All tables within a specific database
  - Table: A specific table in a specific database
  - Column: Specific columns in a table and database

GRANT

- The `GRANT` statement is used to control privileges:
  - `GRANT privileges ( columns ) ON what TO account`
- Example:
  - `GRANT SELECT ON MovieCollection.* TO ‘moviedba’@‘localhost’;`
- This `GRANT` allows “moviedba” connected from “localhost” to perform `SELECT` statements on all tables in the MovieCollection database.

PRIVILEGE LEVELS

- As mentioned previously, privileges can be granted on several different levels. This is achieved with the *what* parameter to the `ON` clause in the `GRANT` statement:
    - `GRANT ALL ON *.* ...`
    - `GRANT ALL ON MovieCollection.* ...`
    - `GRANT ALL ON MovieCollection.movie ...`
    - `GRANT UPDATE ( title ) ON MovieCollection.movie ...`

PRIVILEGE TABLES

- In the mysql database, there are tables which describe the various access controls:
  - `user`: The user accounts and global privileges
  - `db`: Database level privileges
  - `host`: Host level privileges (generally not used)
  - `tables_priv`: Table level privileges
  - `columns_priv`: Column level privileges

LAB

1) Consult the documentation for details on privileges. Read up on the 26 or so privileges available and their meanings. Also, peruse the documentation on the `GRANT` statement.

2) Create a “movieclerk” user which has full select privileges on all tables in MovieCollection, and insert/update/delete privileges on just the actor table in MovieCollection. Test the account.

3) Create a full power “moviedba” user with full access to everything in the MovieCollection database. Test the account.

4) Create a “movieremote” account which has clerk level access from a neighbor machine. Get your neighbor to test the account. Read up on the `mysql` client command to figure out how to do this.

5) Read up on the `REVOKE` command and remove the modification abilities on the clerk accounts. Also, use SET PASSWORD to change the password. Verify.

```
slideshow.end();
```