Jennifer Hoitenga

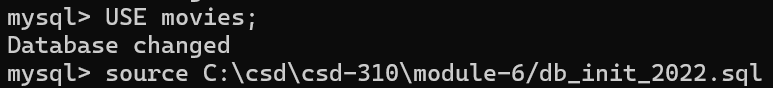
Professor Martinez

January 19th, 2025

Module 6.2 Assignment

Link to GitHub Repo: <https://github.com/jhoitenga/csd-310>

I connected to MySQL using the command line, navigated to the movies database with the USE movies; command and executed the db\_init\_2022.sql script. After running the script, I captured a screenshot of the final output.



A black background with white text

Description automatically generated

To list the tables in a specific database, I first moved to the desired database by using the USE <database name>; command, specifically USE movies;, to switch to the movies database. Then, I executed the SHOW TABLES; command to display a list of all the tables present in the database. After the list of tables appeared, I took a screenshot of the result.

A black screen with white text

Description automatically generated

I ensured the mysql-connector-python driver was installed by running the command pip install mysql-connector-python using Visual Studio Code. I then downloaded the env template file and saved it in the module-6 directory, making sure to rename it with a dot prefix (e.g., .env). This file is excluded from version control because .gitignore typically includes .env files, which helps ensure that sensitive information isn't tracked in Git. Since I used the credentials provided in the db\_init\_2022.sql file, no updates to the .env file were necessary. I installed the python-dotenv package by running pip install python-dotenv, which allows Python to read the .env file and use the stored credentials securely in my scripts. By following these steps, I set up secure handling of MySQL credentials while enabling Python to connect to the database seamlessly.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

I created a Python file named mysql\_test.py and added the provided code. The script starts by importing mysql.connector to connect to MySQL, dotenv to handle the .env file and dotenv\_values to load database credentials securely. Using the .env file, I set up a database configuration object (config) with the user, password, host and database details, plus an extra setting raise\_on\_warnings to help with debugging.

The script uses a try...except...finally block to test the database connection. In the try block, it connects to the database and prints a message confirming the connection, showing the user, host and database. The except block handles errors like invalid credentials or missing databases and prints error messages. The finally block ensures the database connection is closed.

After saving the mysql\_test.py file in the module-6 directory, I ran the script, confirmed the connection worked and took a screenshot of the results.

A computer screen shot of a black screen

Description automatically generated