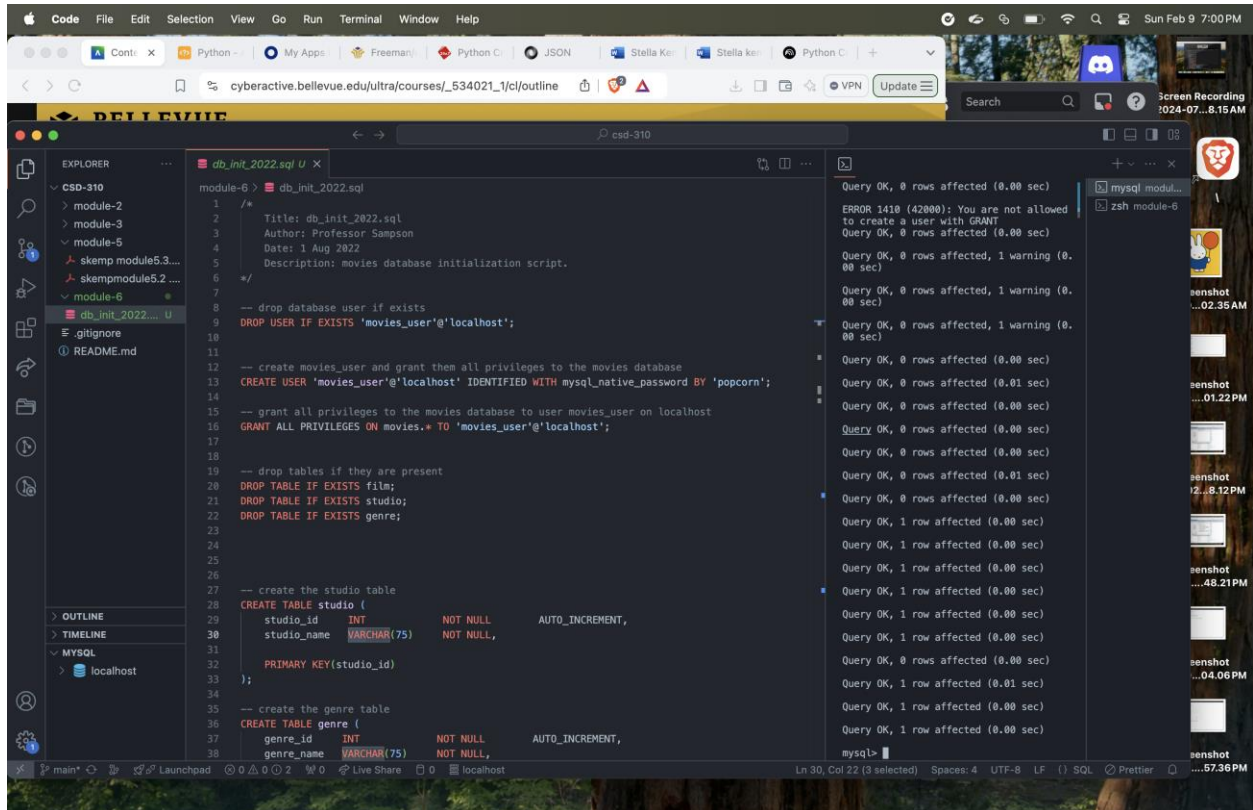


Module 6



Code File Edit Selection View Go Run Terminal Window Help

cyberactive.bellevue.edu/ultra/courses/_534021_1/c/outline

db_init_2022.sql

```
1 /*
2  Title: db_init_2022.sql
3  Author: Professor Sampson
4  Date: 1 Aug 2022
5  Description: movies database initialization script.
6 */
7
8 -- drop database user if exists
9 DROP USER IF EXISTS 'movies_user'@'localhost';
10
11 -- create movies_user and grant them all privileges to
12 CREATE USER 'movies_user'@'localhost' IDENTIFIED WITH
13
14 -- grant all privileges to the movies database to user
15 GRANT ALL PRIVILEGES ON movies.* TO 'movies_user'@'loca
16
17
18 -- drop tables if they are present
19 DROP TABLE IF EXISTS films;
20 DROP TABLE IF EXISTS studio;
21 DROP TABLE IF EXISTS genre;
22
23
24 -- create the studio table
25 CREATE TABLE studio (
26   studio_id INT NOT NULL AUTO_INCREMENT,
27   studio_name VARCHAR(75) NOT NULL,
28   PRIMARY KEY(studio_id)
29 );
30
31 -- create the genre table
32 CREATE TABLE genre (
33   genre_id INT NOT NULL AUTO_INCREMENT,
34   genre_name VARCHAR(75) NOT NULL,
35
36
37
38
```

Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
mysql> SHOW TABLES
->
+-----+
| Tables_in_movies |
+-----+
| film |
| genre |
| studio |
+-----+
3 rows in set (0.00 sec)

mysql>

Code File Edit Selection View Go Run Terminal Window Help

cyberactive.bellevue.edu/ultra/courses/_534021_1/c/outline

mysql_test.py

```
1 from dotenv import dotenv_values
2
3 secrets = dotenv_values(".env")
4
5 #database config object
6 config = {
7     "user": secrets["USER"],
8     "password": secrets["PASSWORD"],
9     "host": secrets["HOST"],
10    "database": secrets["DATABASE"],
11    "raise_on_warnings": True #not in .env file
12 }
13
14 try:
15     # try/catch block for handling potential MySQL database errors
16
17     db = mysql.connector.connect(**config)
18
19     # output the connection status
20     print("\n Database user {} connected to MySQL on host {} with database {}".format(secrets["USER"], secrets["HOST"], secrets["DATABASE"]))
21
22     input("\n\n Press any key to continue...")
23
24 except mysql.connector.Error as err:
25     # "" on error code ""
26
27     if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
28         print(" The supplied username or password are invalid")
29
30     elif err.errno == errorcode.ER_BAD_DB_ERROR:
31         print(" The specified database does not exist")
32
33     else:
34         print(err)
35
36 finally:
37     db.close()
```

Database user root connected to MySQL on host localhost with database movies

Press any key to continue...

