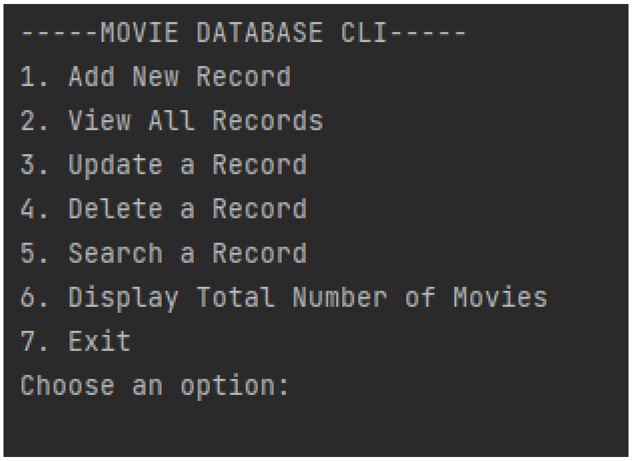
Finals Lab Task 5. CLI using Mysql and Python

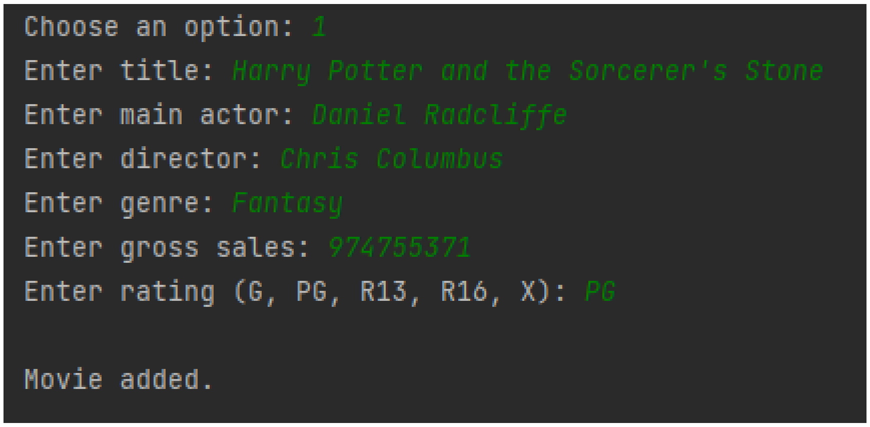
This task involves creating a Python CLI application for managing a `moviesDB` database in MySQL, enabling users to perform CRUD operations through a menu-driven interface; the application should allow users to add, view, update, delete, and search movie records, as well as display the total number of movies stored in the database, with a focus on proper error handling, SQL injection prevention through parameterized queries, and secure database connection management, ultimately requiring the submission of the UI menu, sample output, source code, and an exported SQL file for evaluation.

UI Menu

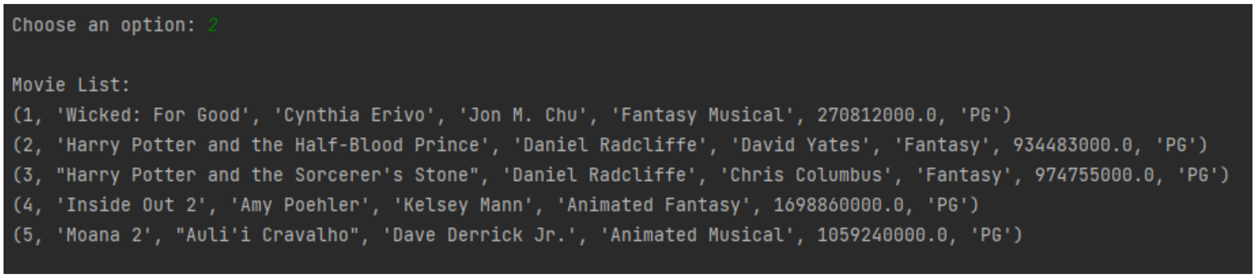


```
-----MOVIE DATABASE CLI-----
1. Add New Record
2. View All Records
3. Update a Record
4. Delete a Record
5. Search a Record
6. Display Total Number of Movies
7. Exit
Choose an option:
```

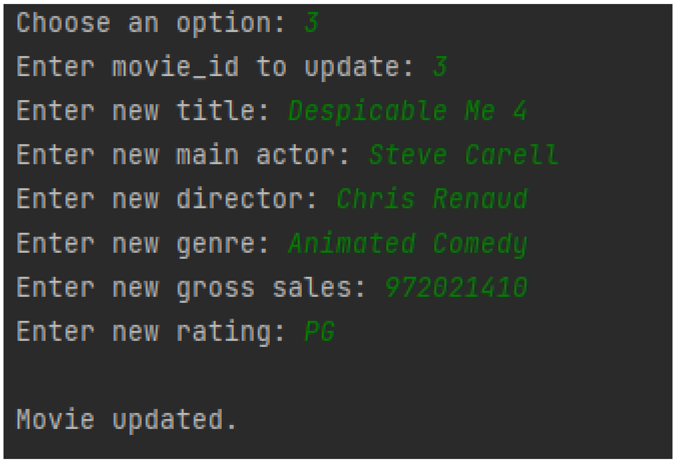
Sample Output



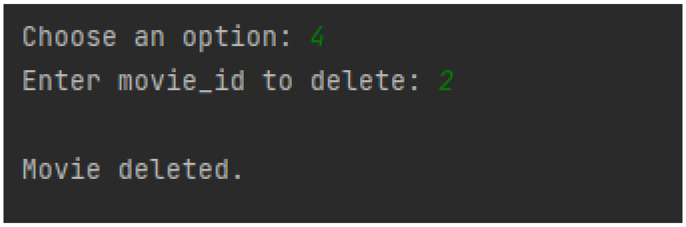
```
Choose an option: 1
Enter title: Harry Potter and the Sorcerer's Stone
Enter main actor: Daniel Radcliffe
Enter director: Chris Columbus
Enter genre: Fantasy
Enter gross sales: 974755371
Enter rating (G, PG, R13, R16, X): PG

Movie added.
```



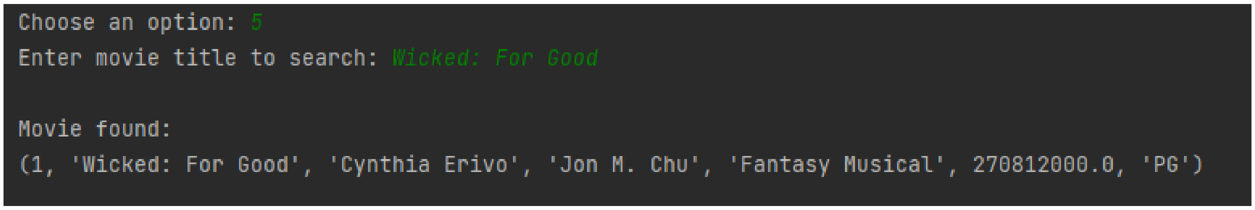
```
Choose an option: 2

Movie List:
(1, 'Wicked: For Good', 'Cynthia Erivo', 'Jon M. Chu', 'Fantasy Musical', 270812000.0, 'PG')
(2, 'Harry Potter and the Half-Blood Prince', 'Daniel Radcliffe', 'David Yates', 'Fantasy', 934483000.0, 'PG')
(3, "Harry Potter and the Sorcerer's Stone", 'Daniel Radcliffe', 'Chris Columbus', 'Fantasy', 974755000.0, 'PG')
(4, 'Inside Out 2', 'Amy Poehler', 'Kelsey Mann', 'Animated Fantasy', 1698860000.0, 'PG')
(5, 'Moana 2', "Auli'i Cravalho", 'Dave Derrick Jr.', 'Animated Musical', 1059240000.0, 'PG')
```



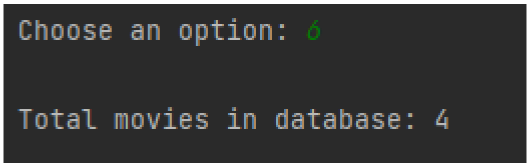
```
Choose an option: 3
Enter movie_id to update: 3
Enter new title: Despicable Me 4
Enter new main actor: Steve Carell
Enter new director: Chris Renaud
Enter new genre: Animated Comedy
Enter new gross sales: 972021410
Enter new rating: PG

Movie updated.
```



```
Choose an option: 4
Enter movie_id to delete: 2

Movie deleted.
```



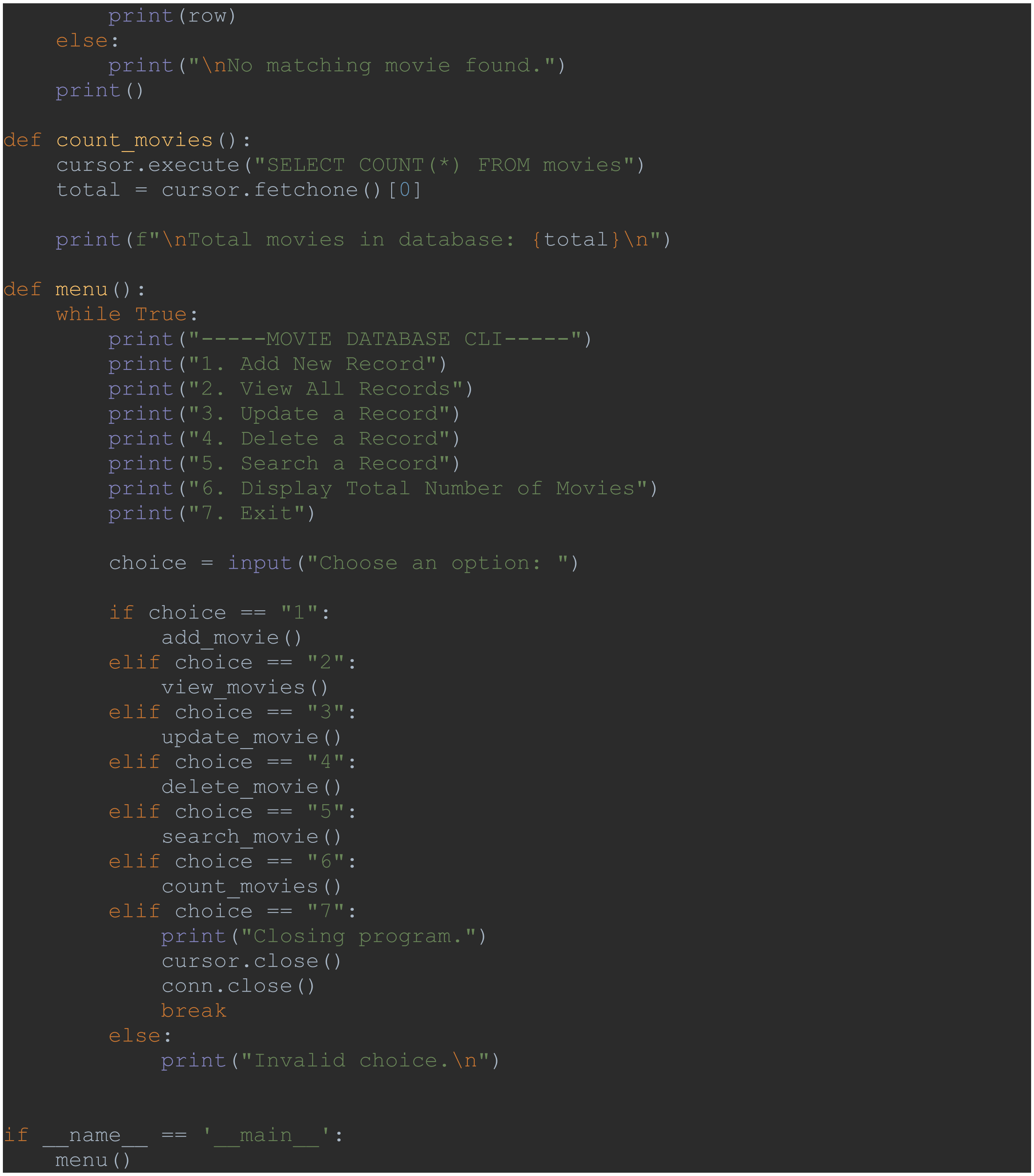
```
Choose an option: 5
Enter movie title to search: Wicked: For Good

Movie found:
(1, 'Wicked: For Good', 'Cynthia Erivo', 'Jon M. Chu', 'Fantasy Musical', 270812000.0, 'PG')
```



```
Choose an option: 6

Total movies in database: 4
```

Source Code

```
import mysql.connector

conn = mysql.connector.connect(
    host="localhost",
    user="tayting05",
    password="Sql2025##",
    database="moviesdb"
)

cursor = conn.cursor()

def add_movie():
    title = input("Enter title: ")
    actor = input("Enter main actor: ")
    director = input("Enter director: ")
    genre = input("Enter genre: ")
    sales = float(input("Enter gross sales: "))
    rating = input("Enter rating (G, PG, R13, R16, X): ")

    cursor.execute(
        "INSERT INTO movies (title, actor, director, genre, sales, ratings)
VALUES (%s, %s, %s, %s, %s, %s)",
        (title, actor, director, genre, sales, rating)
    )

    conn.commit()
    print("\nMovie added.\n")

def view_movies():
    cursor.execute("SELECT * FROM movies")
    rows = cursor.fetchall()

    if rows:
        print("\nMovie List:")
        for r in rows:
            print(r)
    else:
        print("\nNo movies found.")
    print()

def update_movie():
    movie_id = input("Enter movie_id to update: ")
    title = input("Enter new title: ")
    actor = input("Enter new main actor: ")
    director = input("Enter new director: ")
    genre = input("Enter new genre: ")
    sales = float(input("Enter new gross sales: "))
    rating = input("Enter new rating: ")

    cursor.execute(
        "UPDATE movies SET title=%s, actor=%s, director=%s, genre=%s,
sales=%s, ratings=%s WHERE movie_id=%s",
        (title, actor, director, genre, sales, rating, movie_id)
    )

    conn.commit()
    print("\nMovie updated.\n")

def delete_movie():
    movie_id = input("Enter movie_id to delete: ")

    cursor.execute("DELETE FROM movies WHERE movie_id=%s", (movie_id,))
    conn.commit()
    print("\nMovie deleted.\n")

def search_movie():
    keyword = input("Enter movie title to search: ")

    cursor.execute("SELECT * FROM movies WHERE title LIKE %s",
(f"%{keyword}%",))
    row = cursor.fetchone()

    if row:
        print("\nMovie found:")
```



```
        print(row)
    else:
        print("\nNo matching movie found.")
    print()

def count_movies():
    cursor.execute("SELECT COUNT(*) FROM movies")
    total = cursor.fetchone()[0]

    print(f"\nTotal movies in database: {total}\n")

def menu():
    while True:
        print("-----MOVIE DATABASE CLI-----")
        print("1. Add New Record")
        print("2. View All Records")
        print("3. Update a Record")
        print("4. Delete a Record")
        print("5. Search a Record")
        print("6. Display Total Number of Movies")
        print("7. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            add_movie()
        elif choice == "2":
            view_movies()
        elif choice == "3":
            update_movie()
        elif choice == "4":
            delete_movie()
        elif choice == "5":
            search_movie()
        elif choice == "6":
            count_movies()
        elif choice == "7":
            print("Closing program.")
            cursor.close()
            conn.close()
            break
        else:
            print("Invalid choice.\n")


if __name__ == '__main__':
    menu()
```