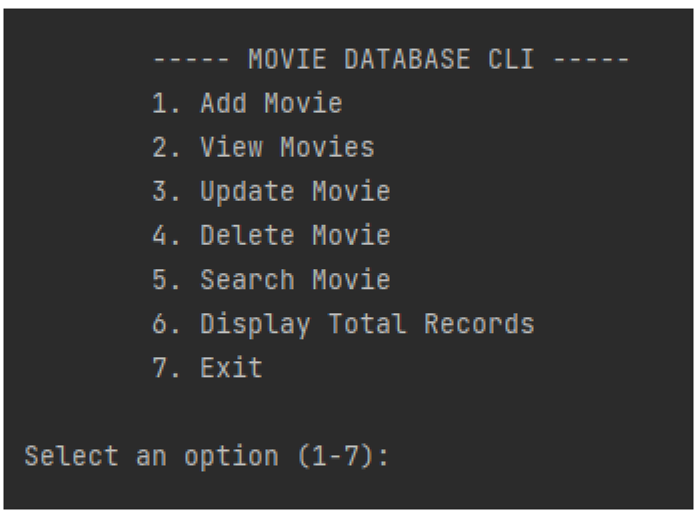
Coronel, Yshian M.
C204

11/29/2025
700P

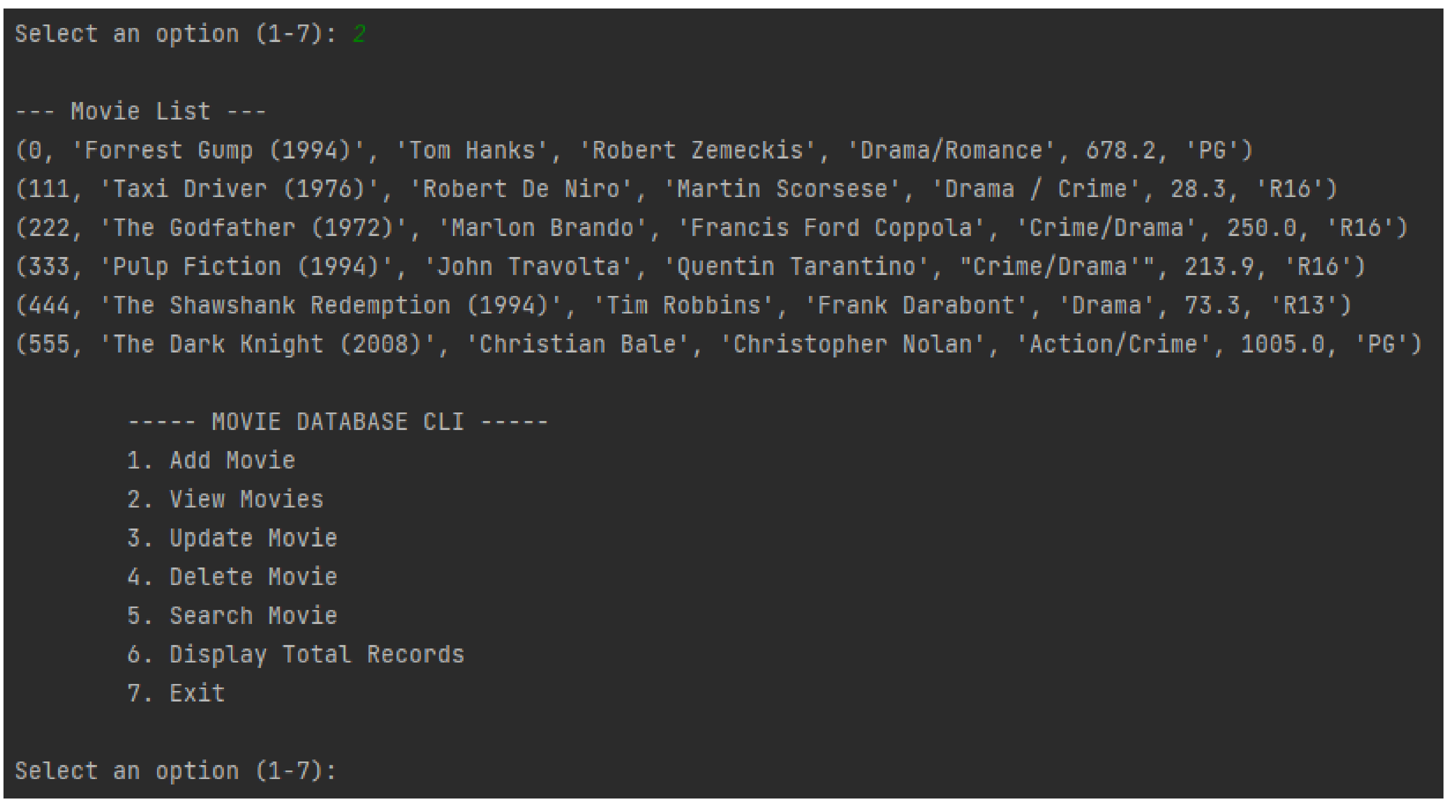
Finals Lab Task 5. CLI using Mysql and Python

UI Menu

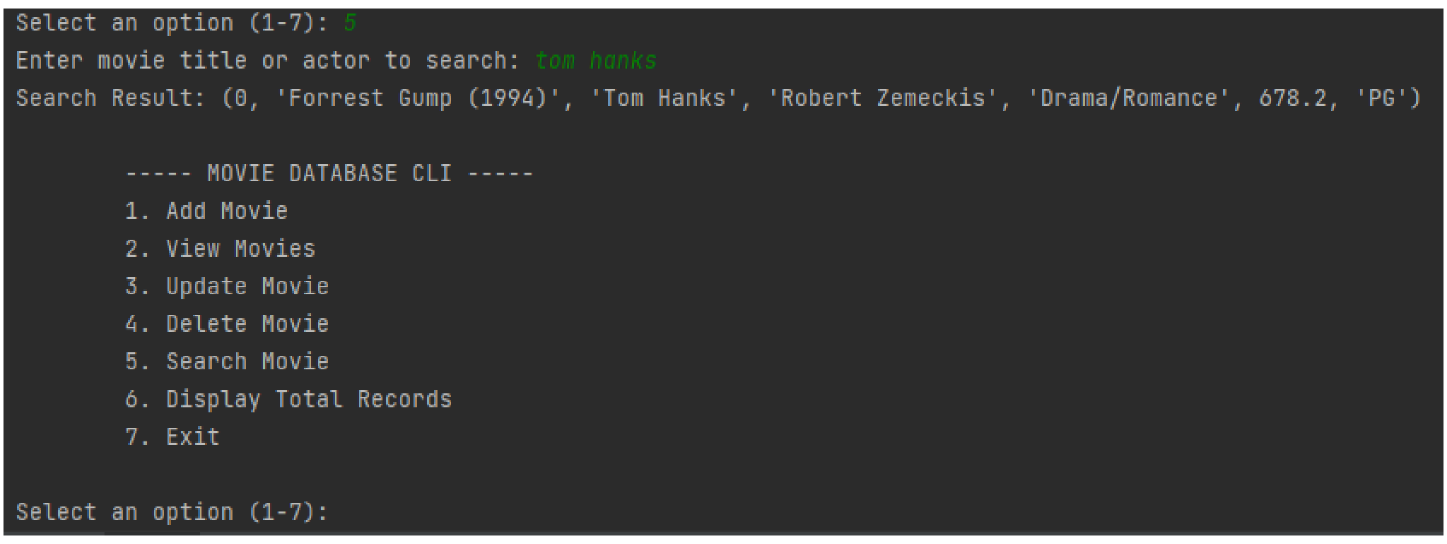


```
        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7):
```

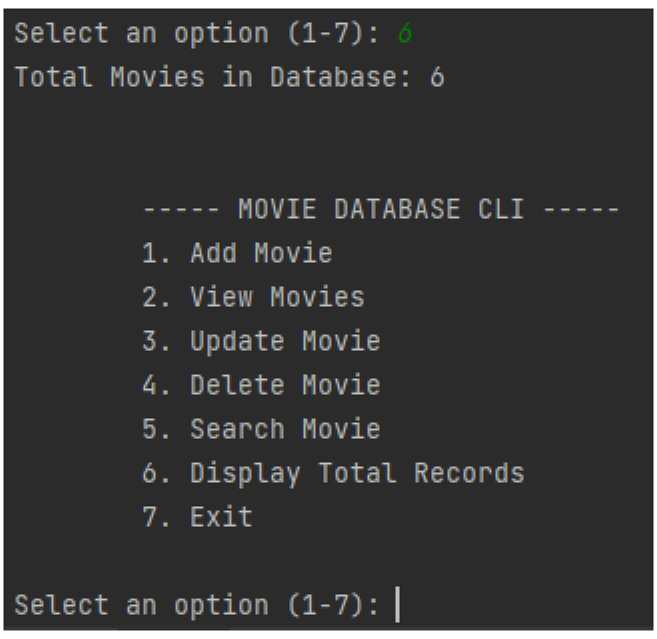
Sample Output



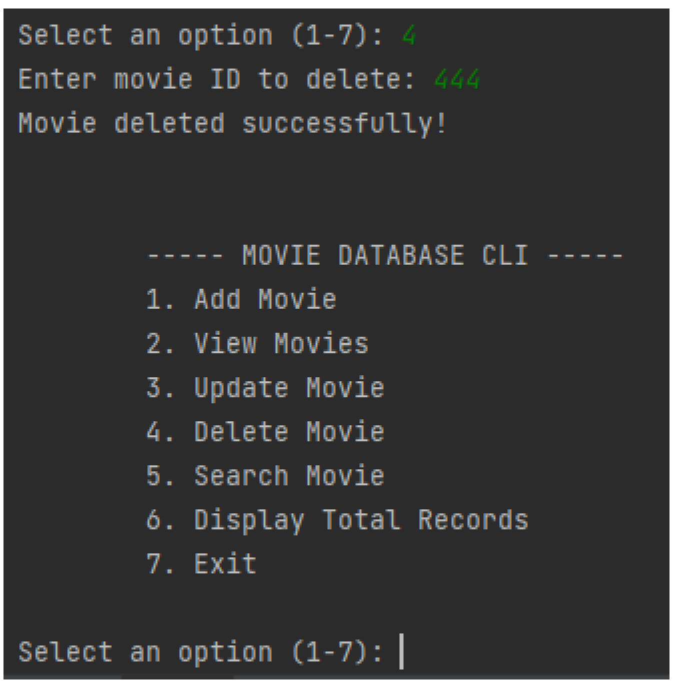
```
Select an option (1-7): 2

--- Movie List ---
(0, 'Forrest Gump (1994)', 'Tom Hanks', 'Robert Zemeckis', 'Drama/Romance', 678.2, 'PG')
(111, 'Taxi Driver (1976)', 'Robert De Niro', 'Martin Scorsese', 'Drama / Crime', 28.3, 'R16')
(222, 'The Godfather (1972)', 'Marlon Brando', 'Francis Ford Coppola', 'Crime/Drama', 250.0, 'R16')
(333, 'Pulp Fiction (1994)', 'John Travolta', 'Quentin Tarantino', "Crime/Drama'", 213.9, 'R16')
(444, 'The Shawshank Redemption (1994)', 'Tim Robbins', 'Frank Darabont', 'Drama', 73.3, 'R13')
(555, 'The Dark Knight (2008)', 'Christian Bale', 'Christopher Nolan', 'Action/Crime', 1005.0, 'PG')

        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7):
```



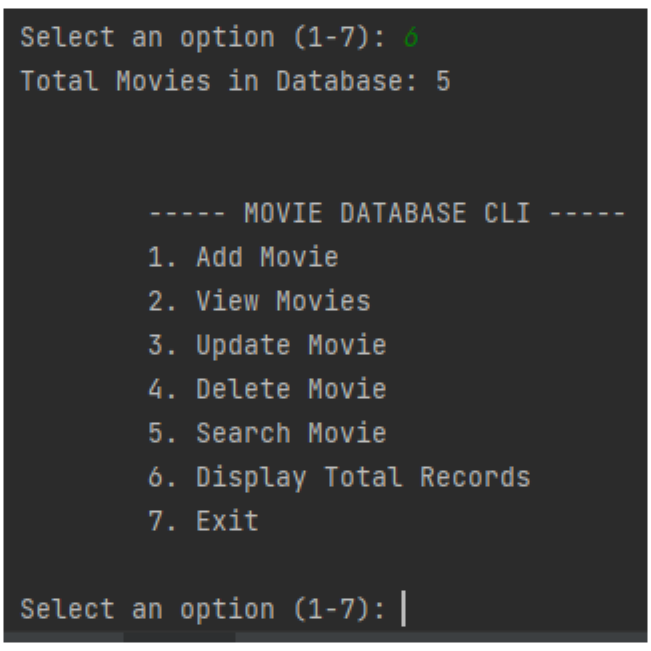
```
Select an option (1-7): 5
Enter movie title or actor to search: tom hanks
Search Result: (0, 'Forrest Gump (1994)', 'Tom Hanks', 'Robert Zemeckis', 'Drama/Romance', 678.2, 'PG')

        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7):
```



```
Select an option (1-7): 6
Total Movies in Database: 6


        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7):
```

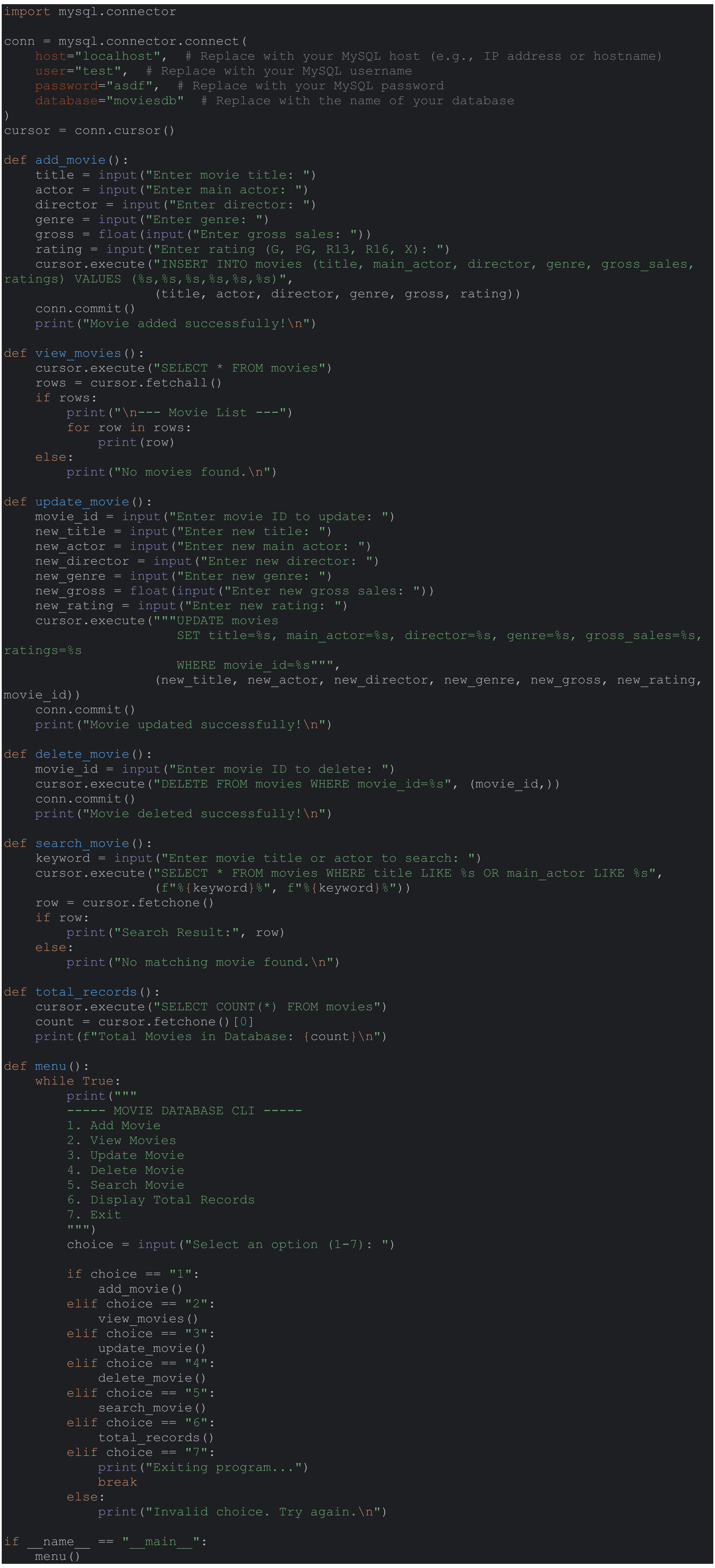


```
Select an option (1-7): 4
Enter movie ID to delete: 444
Movie deleted successfully!


        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7):
```



```
Select an option (1-7): 6
Total Movies in Database: 5


        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit

Select an option (1-7): 
```

TestDb_Demo.py (Source Code)



```python
import mysql.connector

conn = mysql.connector.connect(
    host="localhost",  # Replace with your MySQL host (e.g., IP address or hostname)
    user="test",  # Replace with your MySQL username
    password="asdf",  # Replace with your MySQL password
    database="moviesdb"  # Replace with the name of your database
)
cursor = conn.cursor()

def add_movie():
    title = input("Enter movie title: ")
    actor = input("Enter main actor: ")
    director = input("Enter director: ")
    genre = input("Enter genre: ")
    gross = float(input("Enter gross sales: "))
    rating = input("Enter rating (G, PG, R13, R16, X): ")
    cursor.execute("INSERT INTO movies (title, main_actor, director, genre, gross_sales,
ratings) VALUES (%s,%s,%s,%s,%s,%s)",
                   (title, actor, director, genre, gross, rating))
    conn.commit()
    print("Movie added successfully!\n")

def view_movies():
    cursor.execute("SELECT * FROM movies")
    rows = cursor.fetchall()
    if rows:
        print("\n--- Movie List ---")
        for row in rows:
            print(row)
    else:
        print("No movies found.\n")

def update_movie():
    movie_id = input("Enter movie ID to update: ")
    new_title = input("Enter new title: ")
    new_actor = input("Enter new main actor: ")
    new_director = input("Enter new director: ")
    new_genre = input("Enter new genre: ")
    new_gross = float(input("Enter new gross sales: "))
    new_rating = input("Enter new rating: ")
    cursor.execute("""UPDATE movies
                      SET title=%s, main_actor=%s, director=%s, genre=%s, gross_sales=%s,
ratings=%s
                      WHERE movie_id=%s""",
                   (new_title, new_actor, new_director, new_genre, new_gross, new_rating,
movie_id))
    conn.commit()
    print("Movie updated successfully!\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("Movie deleted successfully!\n")

def search_movie():
    keyword = input("Enter movie title or actor to search: ")
    cursor.execute("SELECT * FROM movies WHERE title LIKE %s OR main_actor LIKE %s",
                   (f"%{keyword}%", f"%{keyword}%"))
    row = cursor.fetchone()
    if row:
        print("Search Result:", row)
    else:
        print("No matching movie found.\n")

def total_records():
    cursor.execute("SELECT COUNT(*) FROM movies")
    count = cursor.fetchone()[0]
    print(f"Total Movies in Database: {count}\n")

def menu():
    while True:
        print("""
        ----- MOVIE DATABASE CLI -----
        1. Add Movie
        2. View Movies
        3. Update Movie
        4. Delete Movie
        5. Search Movie
        6. Display Total Records
        7. Exit
        """)
        choice = input("Select an option (1-7): ")

        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":
            total_records()
        elif choice == "7":
            print("Exiting program...")
            break
        else:
            print("Invalid choice. Try again.\n")

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