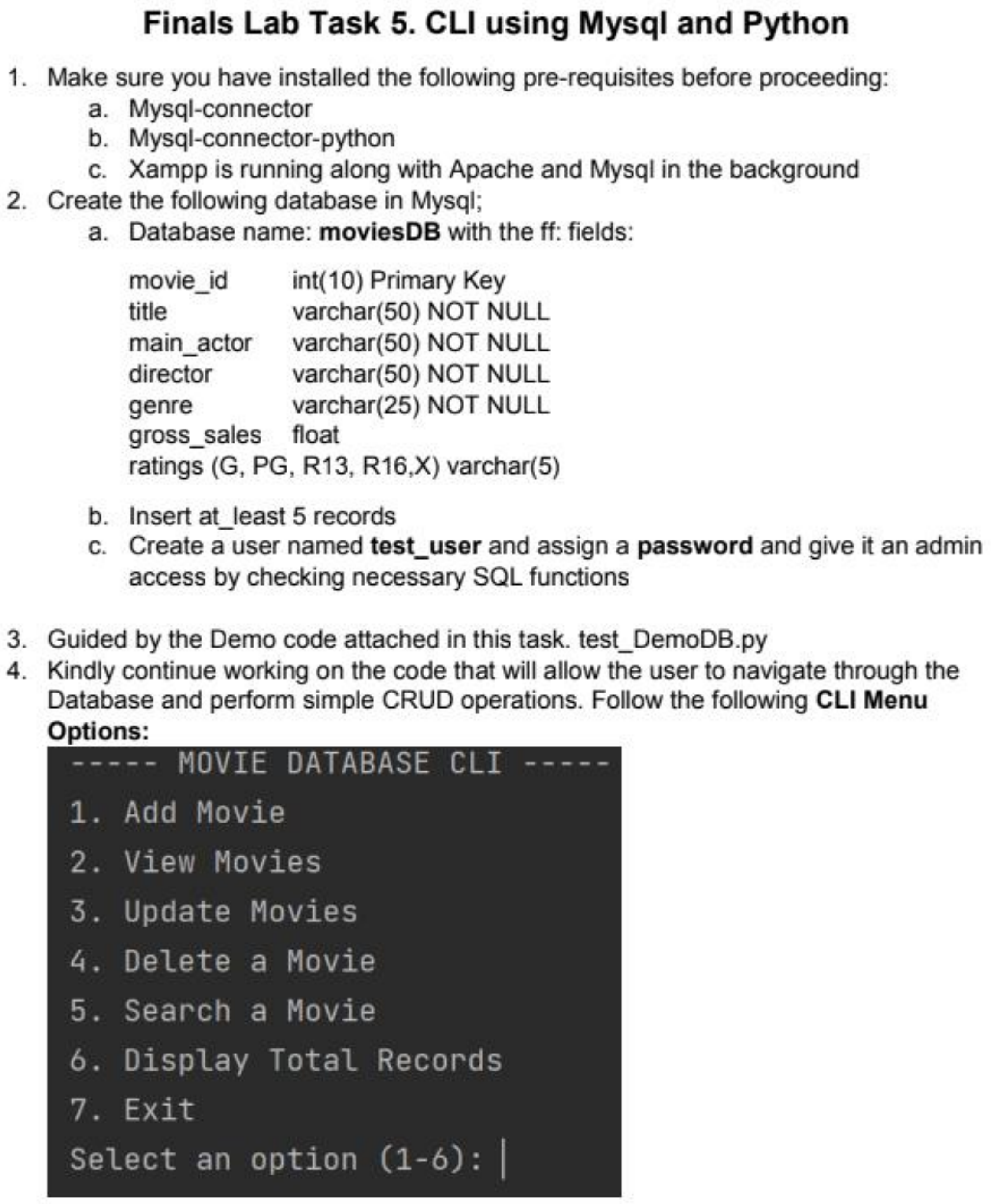
Dumangas, Hannizel December 02, 2025

BSCS – C204 7OOP

Finals Lab Task 5. CRUD CLI using Python and MySQL

Problem



Finals Lab Task 5. CLI using Mysql and Python

1. Make sure you have installed the following pre-requisites before proceeding:
   a. Mysql-connector
   b. Mysql-connector-python
   c. Xampp is running along with Apache and Mysql in the background
2. Create the following database in Mysql;
   a. Database name: **moviesDB** with the ff: fields:

movie_id	int(10) Primary Key
title	varchar(50) NOT NULL
main_actor	varchar(50) NOT NULL
director	varchar(50) NOT NULL
genre	varchar(25) NOT NULL
gross_sales	float
ratings (G, PG, R13, R16,X) varchar(5)	

   b. Insert at_least 5 records
   c. Create a user named **test_user** and assign a **password** and give it an admin access by checking necessary SQL functions

3. Guided by the Demo code attached in this task. test_DemoDB.py
4. Kindly continue working on the code that will allow the user to navigate through the Database and perform simple CRUD operations. Follow the following **CLI Menu Options:**

```
----- MOVIE DATABASE CLI -----
1. Add Movie
2. View Movies
3. Update Movies
4. Delete a Movie
5. Search a Movie
6. Display Total Records
7. Exit
Select an option (1-6): |
```

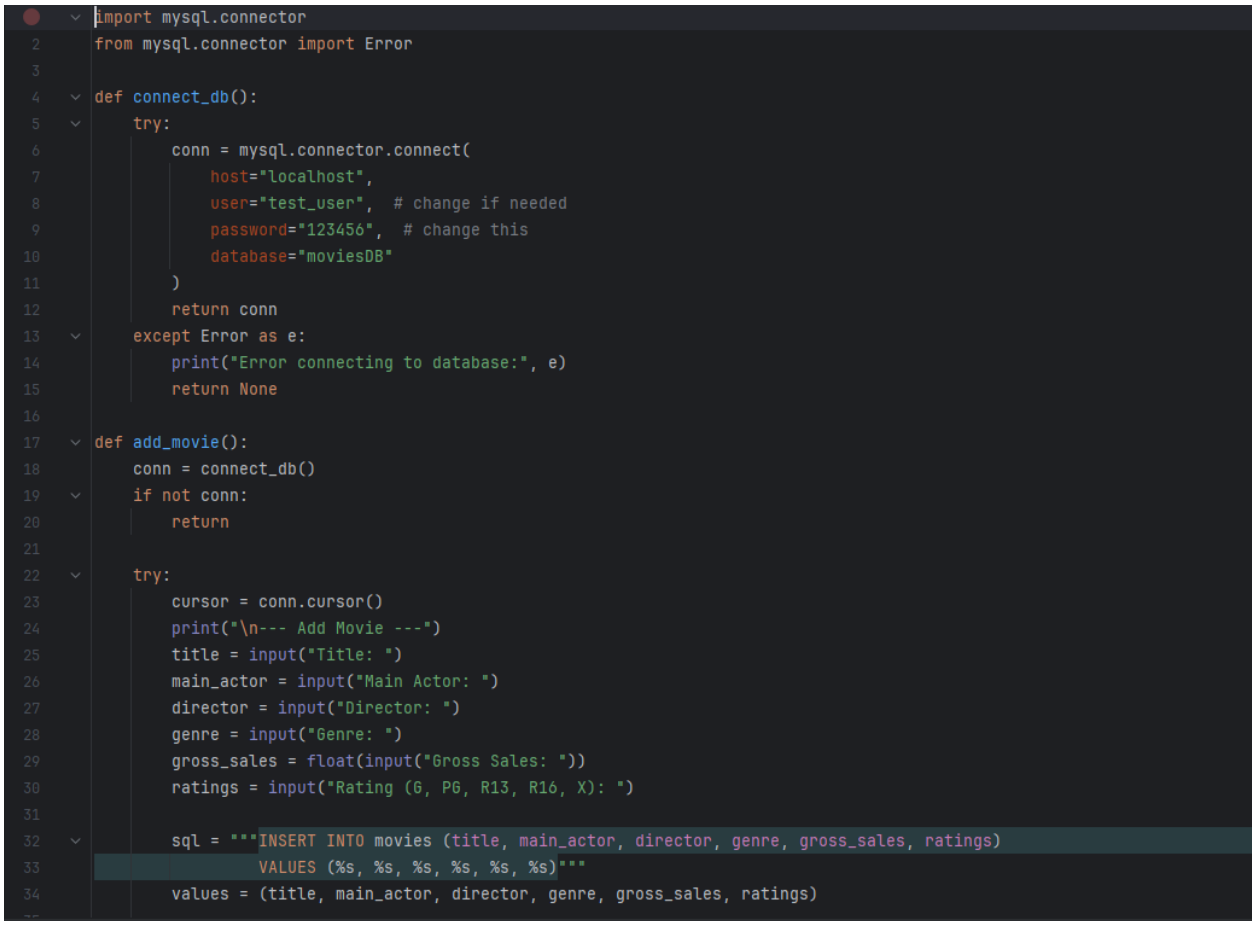
5. The user should be able perform the ff: in your program.

MOVIE DATABASE CRUD APP

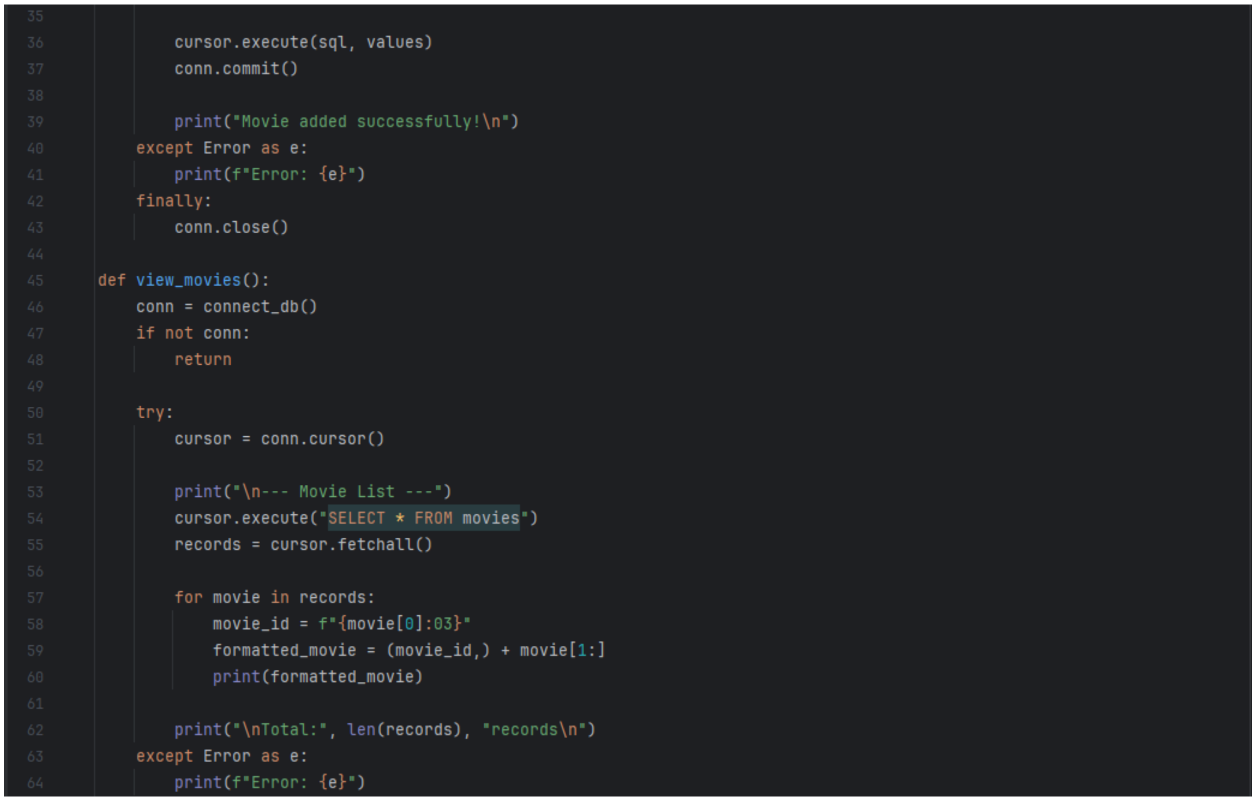
1- Add New Record
2- View all records,
3- Update a Record and show the updates,
4- Delete a record
5- Search A Record
6- Display **Total Numbers** of Movies stored in the database
7- Exit

6. For additional challenge, Task – You are to add a **SEARCH option** in the MENU that will allow the user to search by Name or emp_id, then display the information about the record being search. You may use Like statement and fetchOne method in my SQL to do this,
7. You are also going to add a method that will display the the **total number of records** in your database – You may use SQL count statement for this.
8. What to submit:
    a. UI Menu
    b. Sample Output
    c. Source Code
    d. Exported sql file

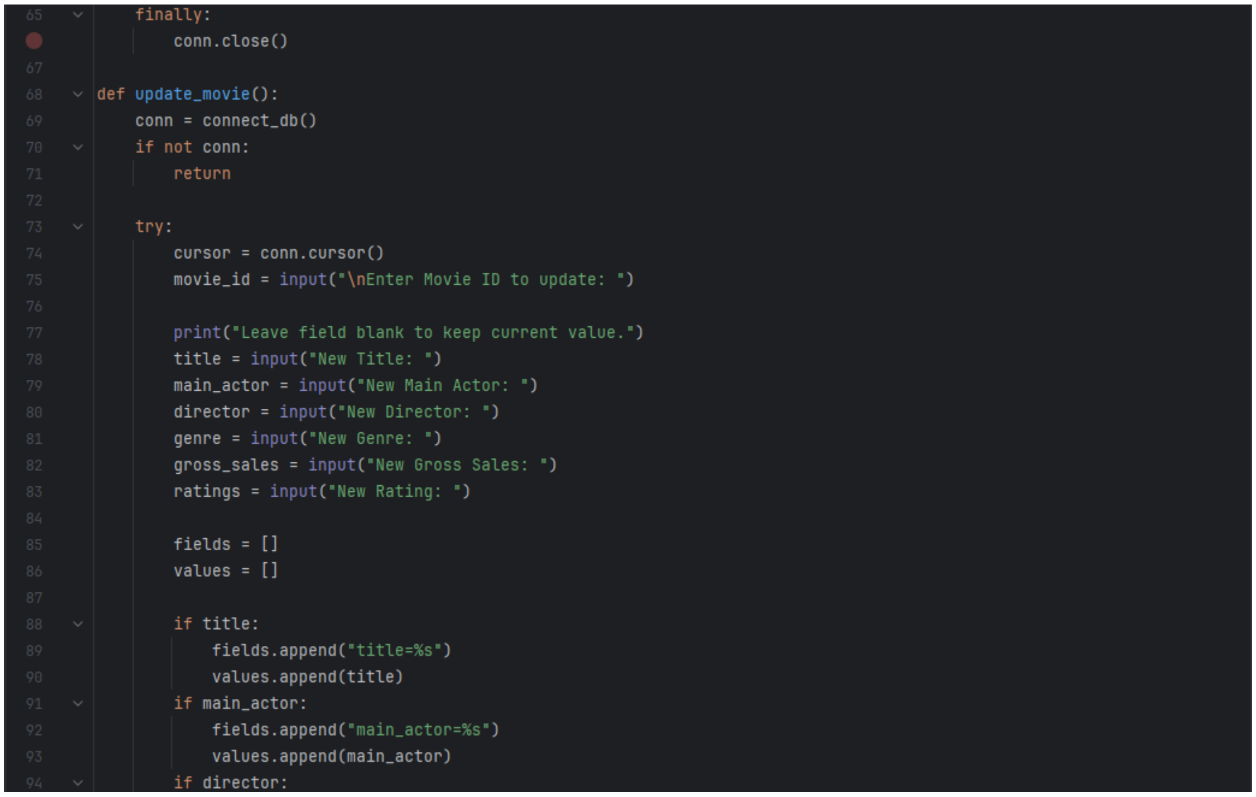
Source Code



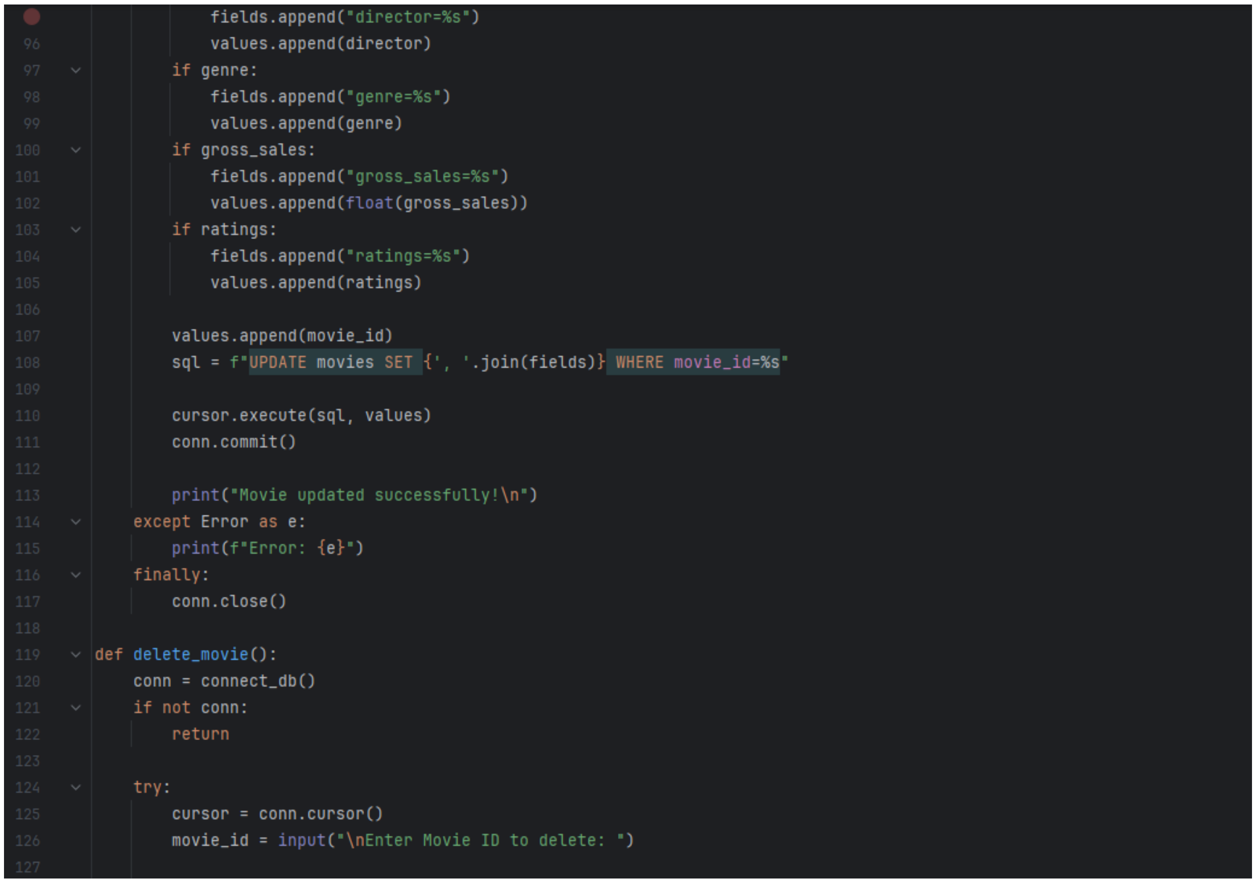
```python
import mysql.connector
from mysql.connector import Error

def connect_db():
    try:
        conn = mysql.connector.connect(
            host="localhost",
            user="test_user",  # change if needed
            password="123456",  # change this
            database="moviesDB"
        )
        return conn
    except Error as e:
        print("Error connecting to database:", e)
        return None

def add_movie():
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()
        print("\n--- Add Movie ---")
        title = input("Title: ")
        main_actor = input("Main Actor: ")
        director = input("Director: ")
        genre = input("Genre: ")
        gross_sales = float(input("Gross Sales: "))
        ratings = input("Rating (G, PG, R13, R16, X): ")

        sql = """INSERT INTO movies (title, main_actor, director, genre, gross_sales, ratings)
                 VALUES (%s, %s, %s, %s, %s, %s)"""
        values = (title, main_actor, director, genre, gross_sales, ratings)
```



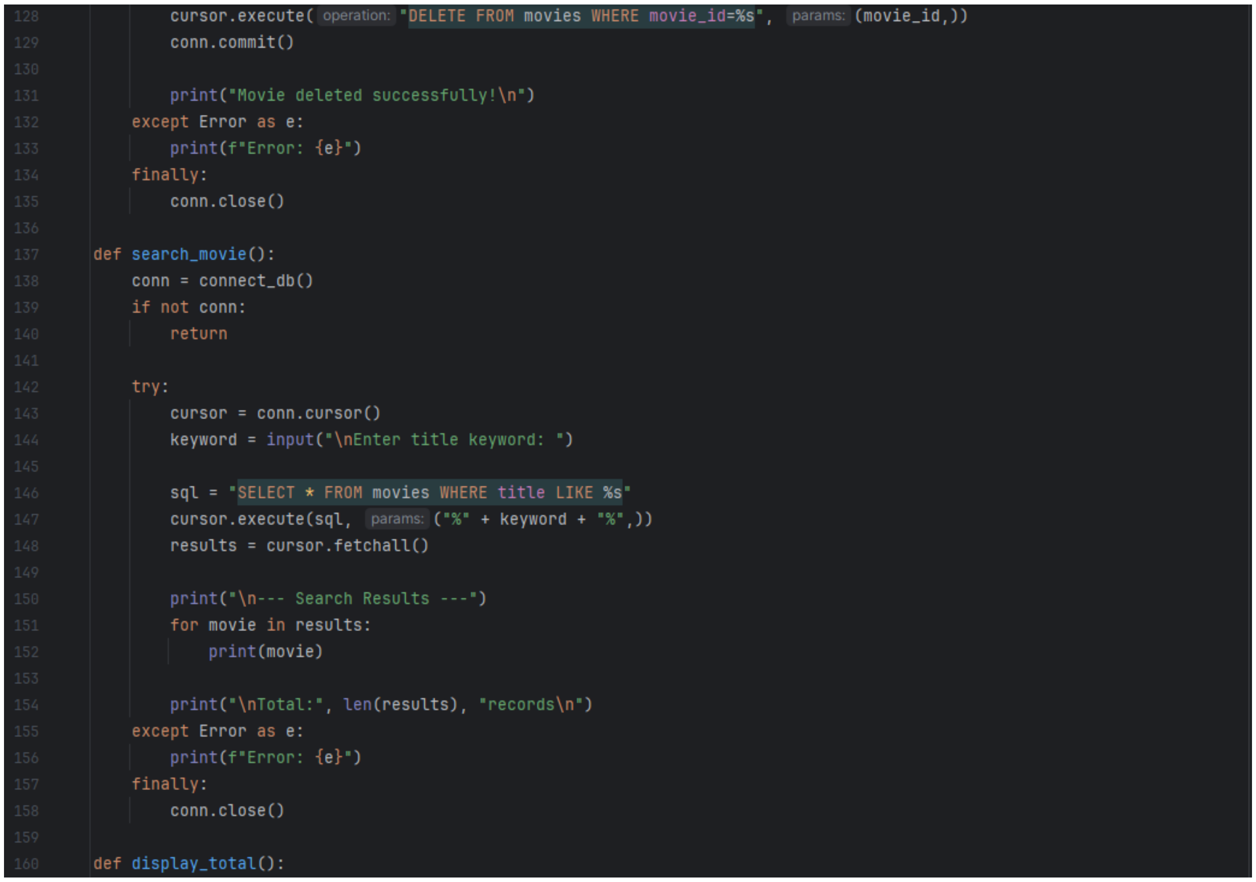
```python

        cursor.execute(sql, values)
        conn.commit()

        print("Movie added successfully!\n")
    except Error as e:
        print(f"Error: {e}")
    finally:
        conn.close()

def view_movies():
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()

        print("\n--- Movie List ---")
        cursor.execute("SELECT * FROM movies")
        records = cursor.fetchall()

        for movie in records:
            movie_id = f"{movie[0]:03}"
            formatted_movie = (movie_id,) + movie[1:]
            print(formatted_movie)

        print("\nTotal:", len(records), "records\n")
    except Error as e:
        print(f"Error: {e}")
```



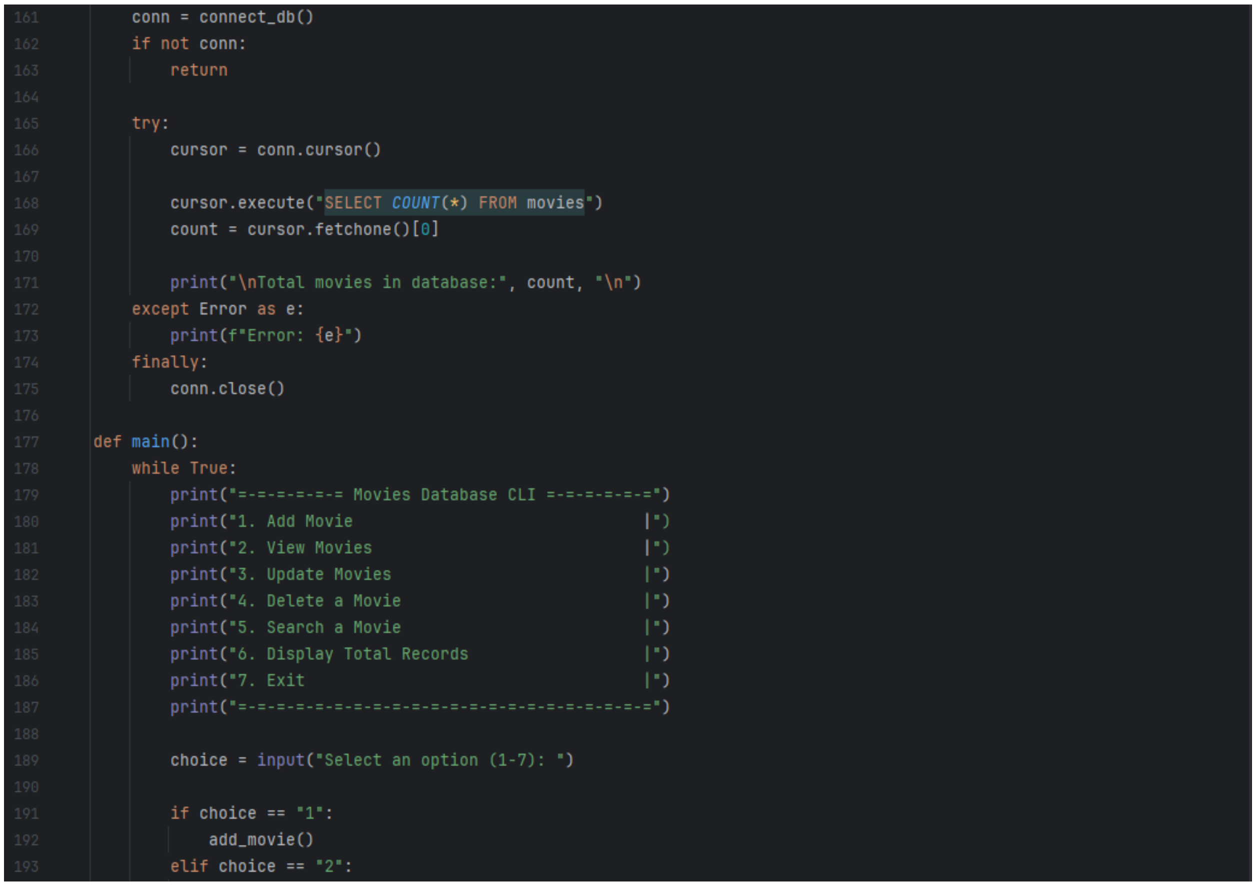
```python
    finally:
        conn.close()

def update_movie():
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()
        movie_id = input("\nEnter Movie ID to update: ")

        print("Leave field blank to keep current value.")
        title = input("New Title: ")
        main_actor = input("New Main Actor: ")
        director = input("New Director: ")
        genre = input("New Genre: ")
        gross_sales = input("New Gross Sales: ")
        ratings = input("New Rating: ")

        fields = []
        values = []

        if title:
            fields.append("title=%s")
            values.append(title)
        if main_actor:
            fields.append("main_actor=%s")
            values.append(main_actor)
        if director:
```



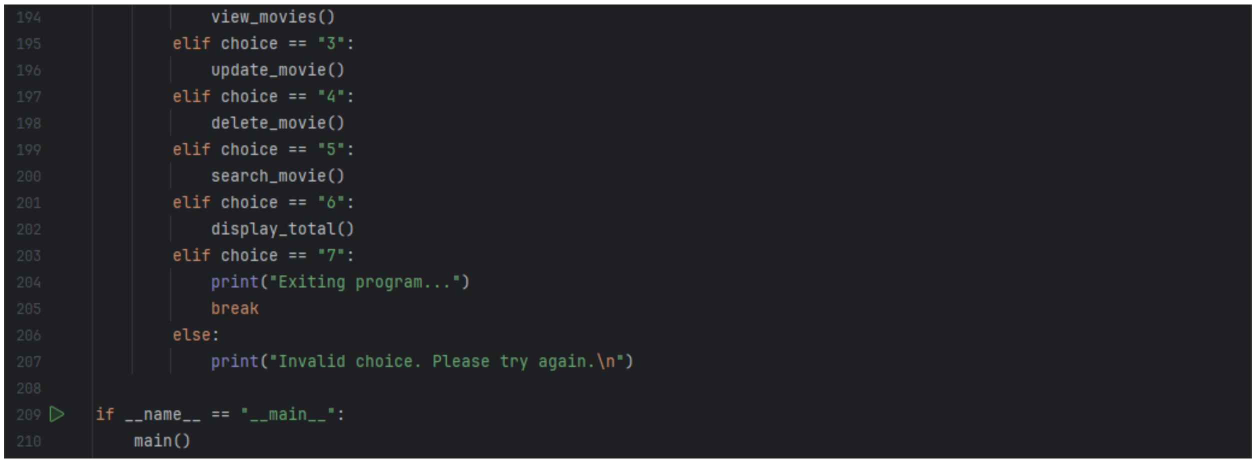
```python
            fields.append("director=%s")
            values.append(director)
        if genre:
            fields.append("genre=%s")
            values.append(genre)
        if gross_sales:
            fields.append("gross_sales=%s")
            values.append(float(gross_sales))
        if ratings:
            fields.append("ratings=%s")
            values.append(ratings)

        values.append(movie_id)
        sql = f"UPDATE movies SET {', '.join(fields)} WHERE movie_id=%s"

        cursor.execute(sql, values)
        conn.commit()

        print("Movie updated successfully!\n")
    except Error as e:
        print(f"Error: {e}")
    finally:
        conn.close()

def delete_movie():
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()
        movie_id = input("\nEnter Movie ID to delete: ")

```



```python
        cursor.execute( operation: "DELETE FROM movies WHERE movie_id=%s",  params: (movie_id,))
        conn.commit()

        print("Movie deleted successfully!\n")
    except Error as e:
        print(f"Error: {e}")
    finally:
        conn.close()

def search_movie():
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()
        keyword = input("\nEnter title keyword: ")

        sql = "SELECT * FROM movies WHERE title LIKE %s"
        cursor.execute(sql,  params: ("%" + keyword + "%",))
        results = cursor.fetchall()

        print("\n--- Search Results ---")
        for movie in results:
            print(movie)

        print("\nTotal:", len(results), "records\n")
    except Error as e:
        print(f"Error: {e}")
    finally:
        conn.close()

def display_total():
```

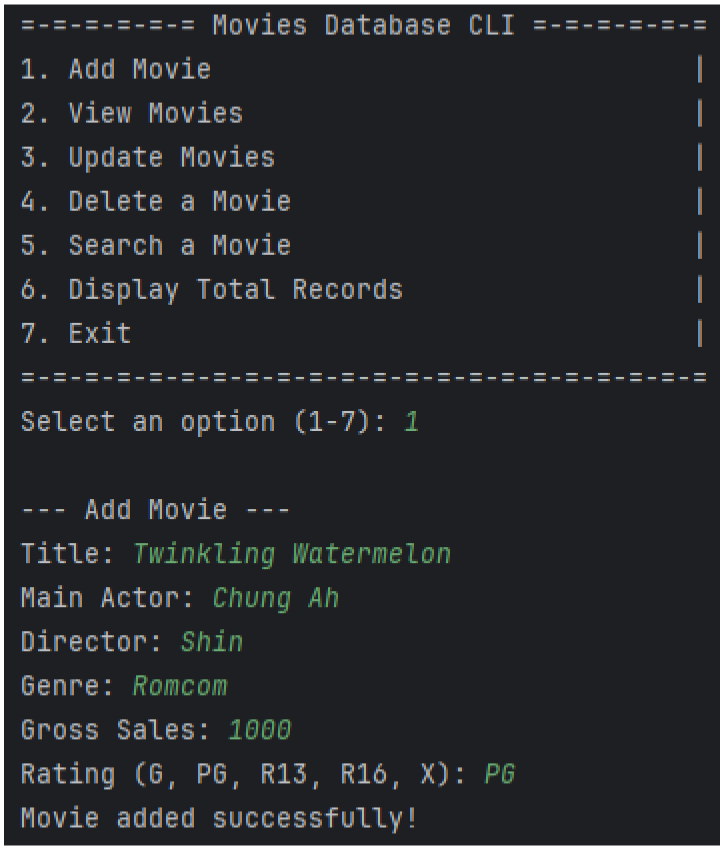


```python
    conn = connect_db()
    if not conn:
        return

    try:
        cursor = conn.cursor()

        cursor.execute("SELECT COUNT(*) FROM movies")
        count = cursor.fetchone()[0]

        print("\nTotal movies in database:", count, "\n")
    except Error as e:
        print(f"Error: {e}")
    finally:
        conn.close()

def main():
    while True:
        print("=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=")
        print("1. Add Movie                              |")
        print("2. View Movies                            |")
        print("3. Update Movies                          |")
        print("4. Delete a Movie                         |")
        print("5. Search a Movie                         |")
        print("6. Display Total Records                  |")
        print("7. Exit                                   |")
        print("=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=")

        choice = input("Select an option (1-7): ")

        if choice == "1":
            add_movie()
        elif choice == "2":
```

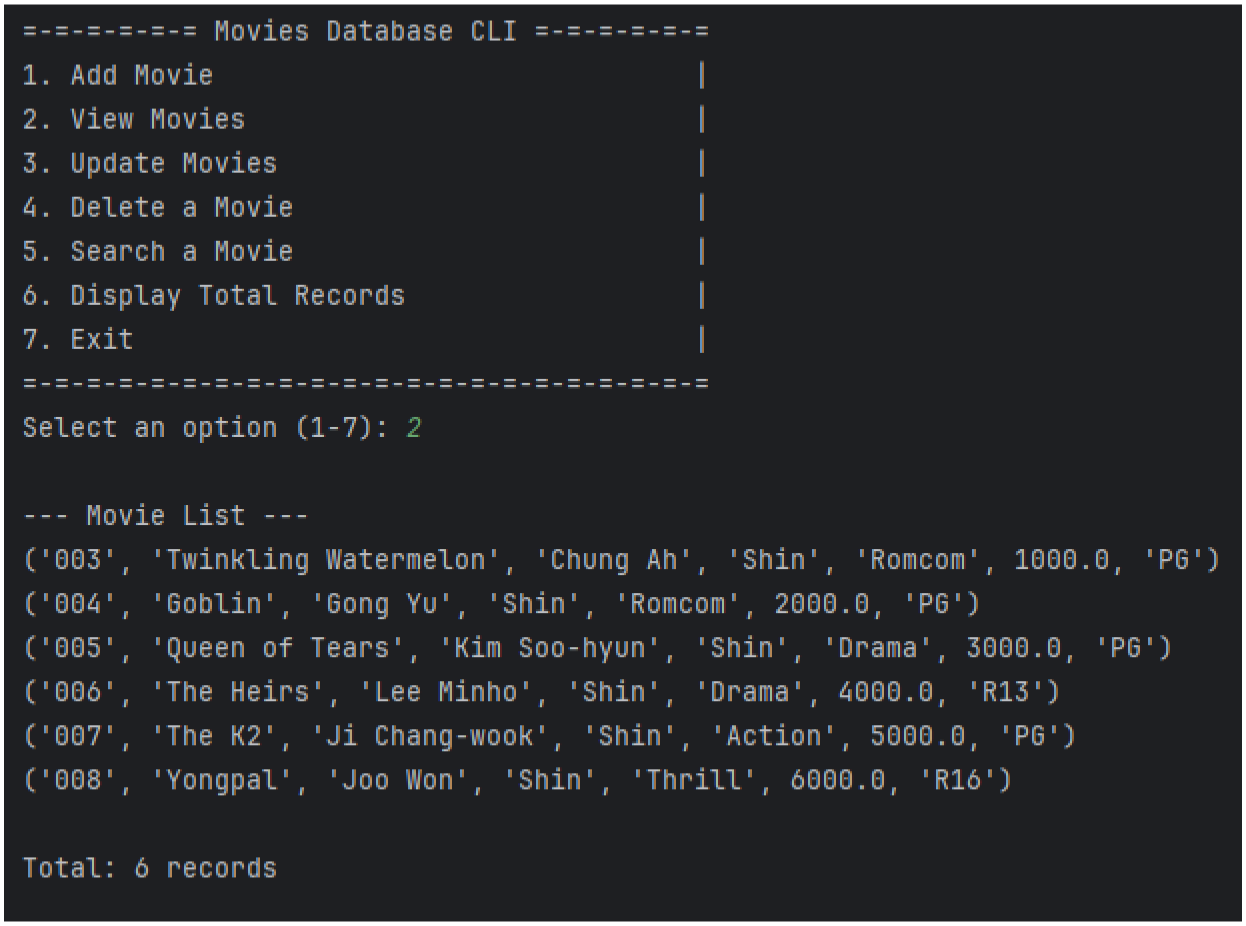


```python
            view_movies()
        elif choice == "3":
            update_movie()
        elif choice == "4":
            delete_movie()
        elif choice == "5":
            search_movie()
        elif choice == "6":
            display_total()
        elif choice == "7":
            print("Exiting program...")
            break
        else:
            print("Invalid choice. Please try again.\n")

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

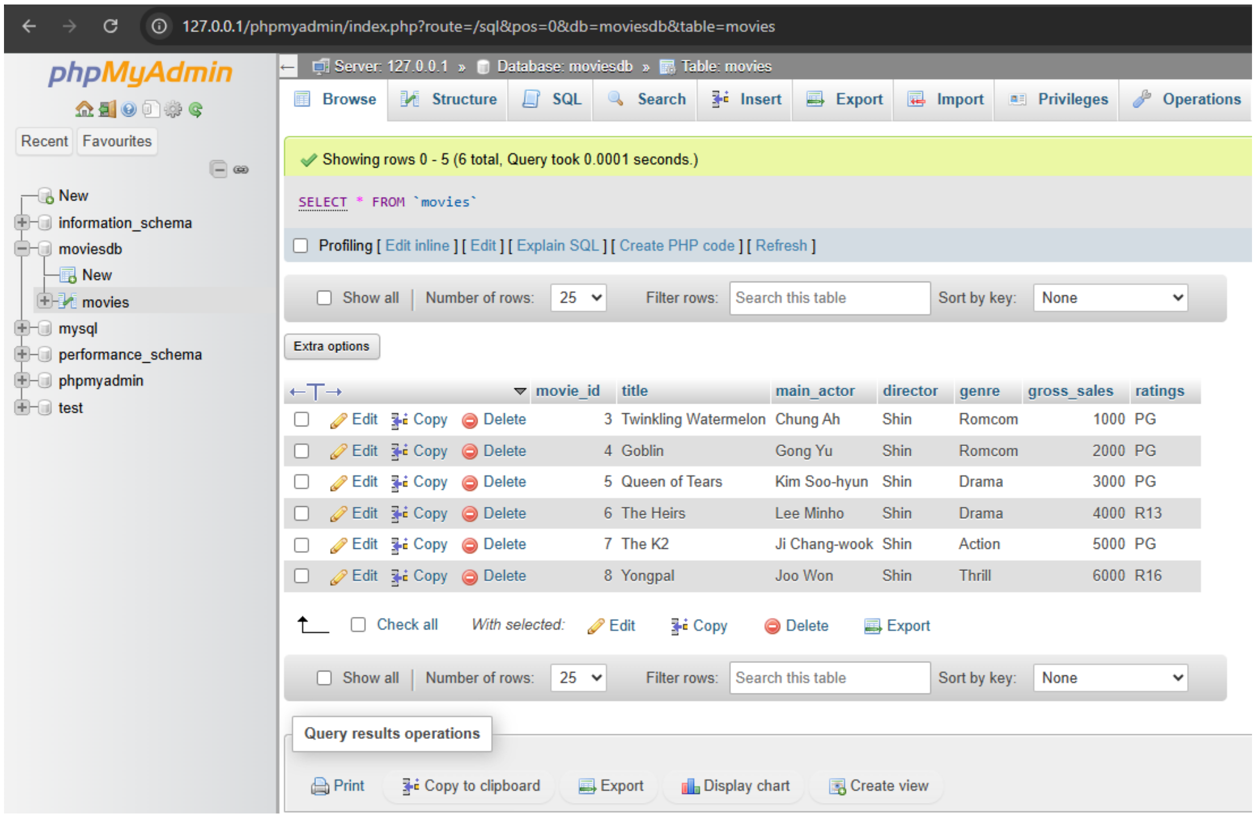
Sample Output



```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 1

--- Add Movie ---
Title: Twinkling Watermelon
Main Actor: Chung Ah
Director: Shin
Genre: Romcom
Gross Sales: 1000
Rating (G, PG, R13, R16, X): PG
Movie added successfully!
```



```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 2

--- Movie List ---
('003', 'Twinkling Watermelon', 'Chung Ah', 'Shin', 'Romcom', 1000.0, 'PG')
('004', 'Goblin', 'Gong Yu', 'Shin', 'Romcom', 2000.0, 'PG')
('005', 'Queen of Tears', 'Kim Soo-hyun', 'Shin', 'Drama', 3000.0, 'PG')
('006', 'The Heirs', 'Lee Minho', 'Shin', 'Drama', 4000.0, 'R13')
('007', 'The K2', 'Ji Chang-wook', 'Shin', 'Action', 5000.0, 'PG')
('008', 'Yongpal', 'Joo Won', 'Shin', 'Thrill', 6000.0, 'R16')

Total: 6 records
```

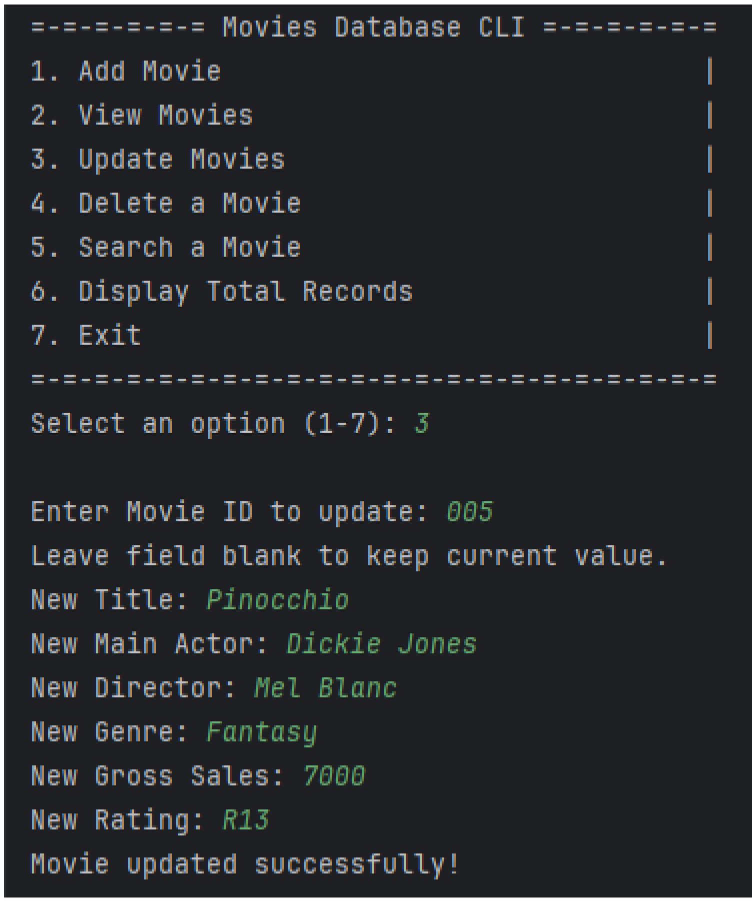

127.0.0.1/phpmyadmin/index.php?route=/sql&pos=0&db=moviesdb&table=movies
phpMyAdmin
Recent
Favourites
New
information_schema
moviesdb
New
movies
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: moviesdb » Table: movies
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations

Showing rows 0 - 5 (6 total, Query took 0.0001 seconds.)

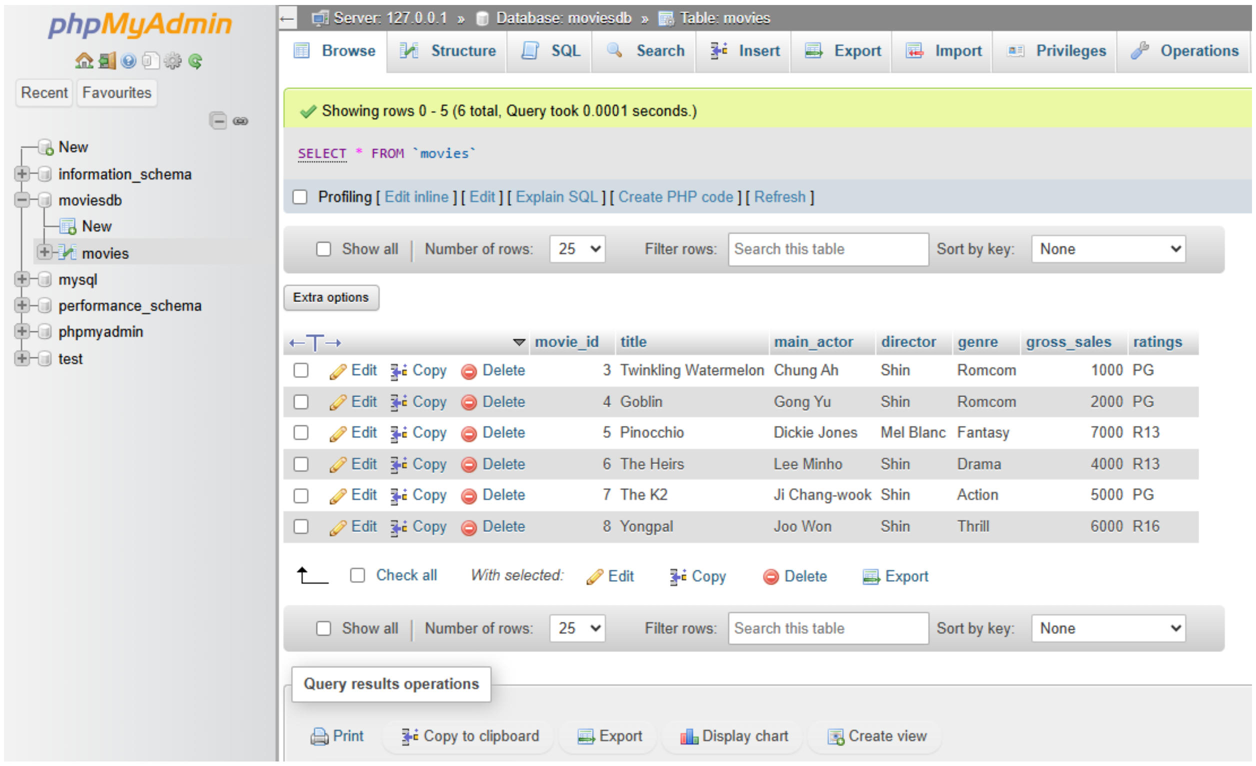
```
SELECT * FROM `movies`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None
Extra options

movie_id	title	main_actor	director	genre	gross_sales	ratings
3	Twinkling Watermelon	Chung Ah	Shin	Romcom	1000	PG
4	Goblin	Gong Yu	Shin	Romcom	2000	PG
5	Queen of Tears	Kim Soo-hyun	Shin	Drama	3000	PG
6	The Heirs	Lee Minho	Shin	Drama	4000	R13
7	The K2	Ji Chang-wook	Shin	Action	5000	PG
8	Yongpal	Joo Won	Shin	Thrill	6000	R16



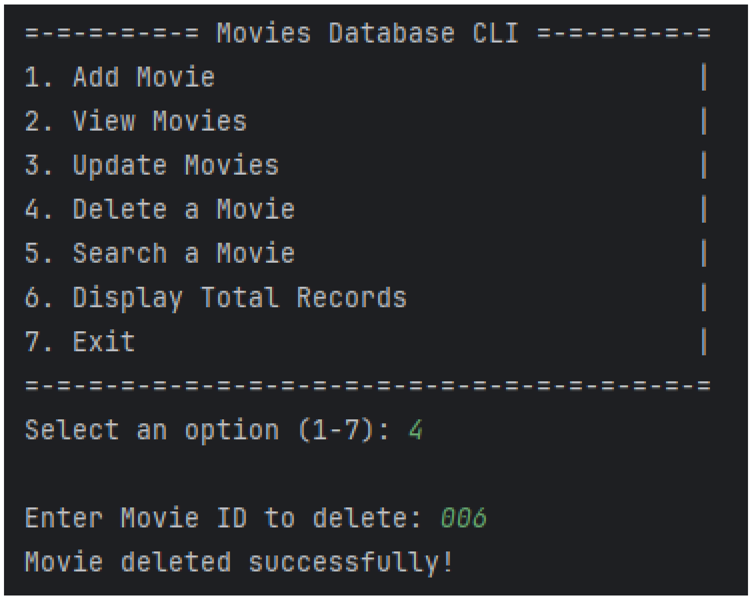
```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 3

Enter Movie ID to update: 005
Leave field blank to keep current value.
New Title: Pinocchio
New Main Actor: Dickie Jones
New Director: Mel Blanc
New Genre: Fantasy
New Gross Sales: 7000
New Rating: R13
Movie updated successfully!
```



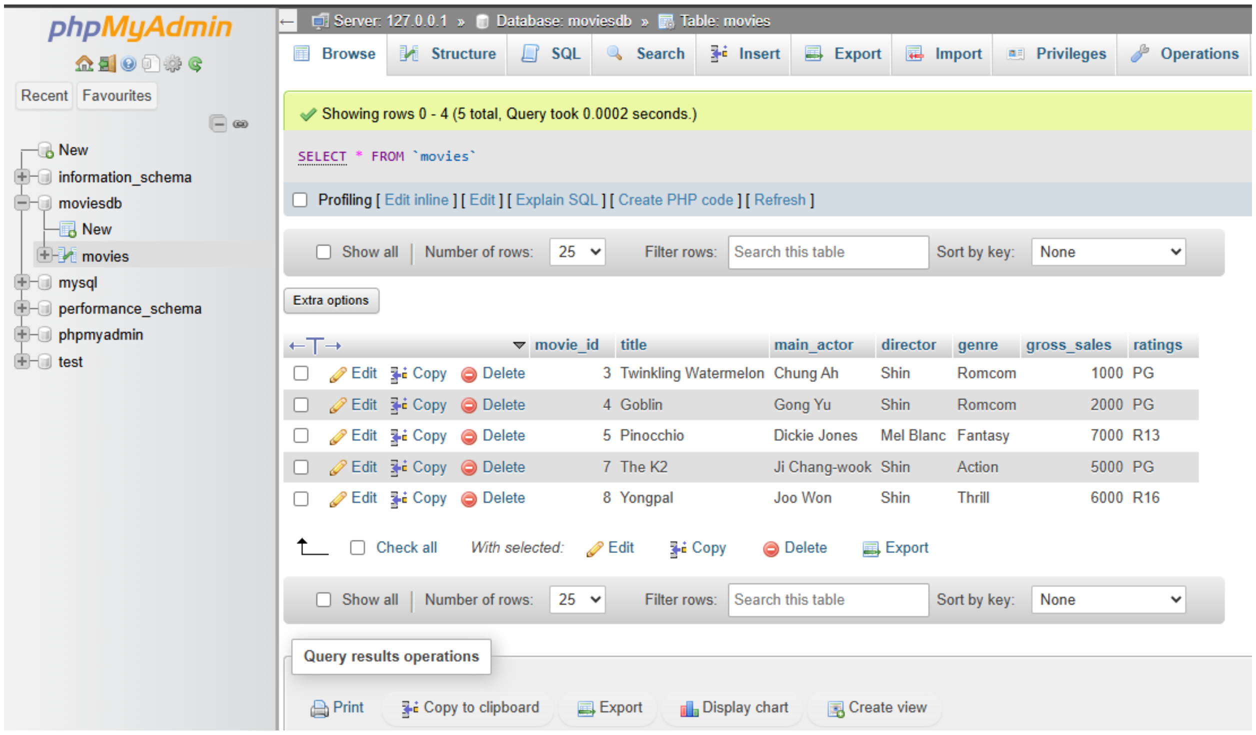
Showing rows 0 - 5 (6 total, Query took 0.0001 seconds.)

```
SELECT * FROM `movies`
```

movie_id	title	main_actor	director	genre	gross_sales	ratings
3	Twinkling Watermelon	Chung Ah	Shin	Romcom	1000	PG
4	Goblin	Gong Yu	Shin	Romcom	2000	PG
5	Pinocchio	Dickie Jones	Mel Blanc	Fantasy	7000	R13
6	The Heirs	Lee Minho	Shin	Drama	4000	R13
7	The K2	Ji Chang-wook	Shin	Action	5000	PG
8	Yongpal	Joo Won	Shin	Thrill	6000	R16



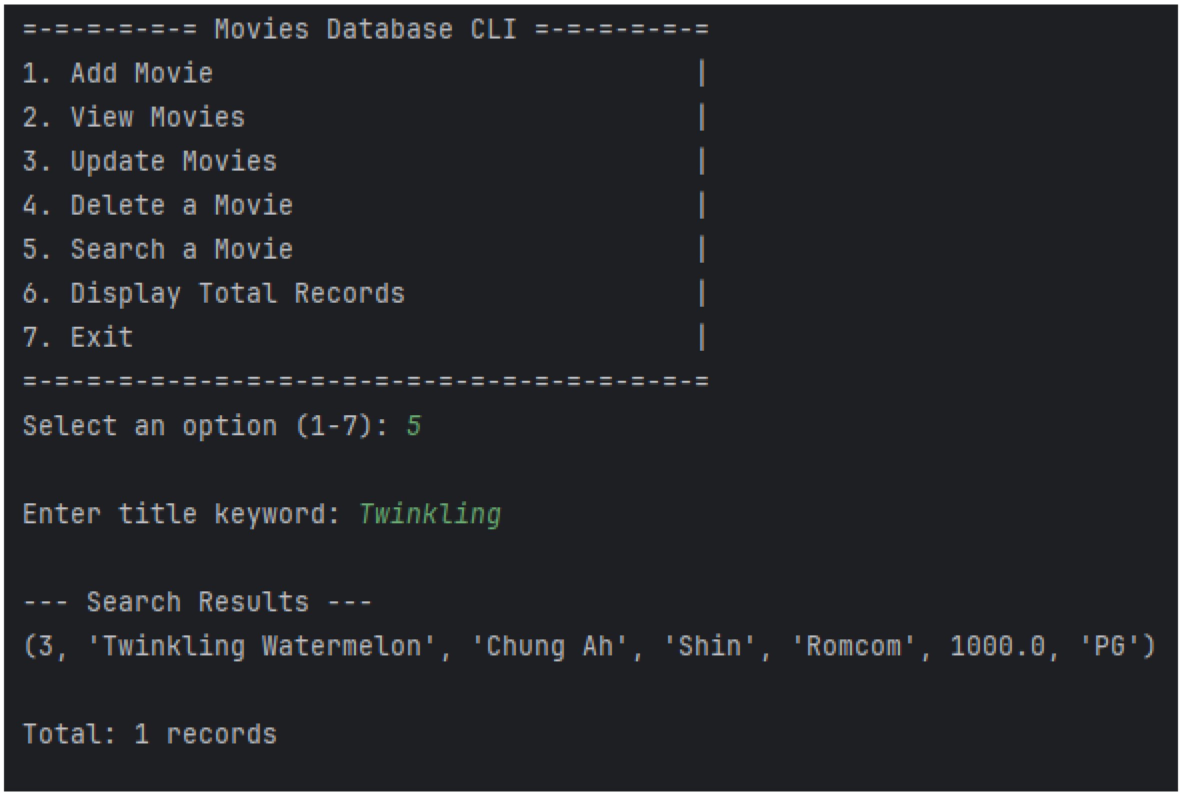
```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 4

Enter Movie ID to delete: 006
Movie deleted successfully!
```



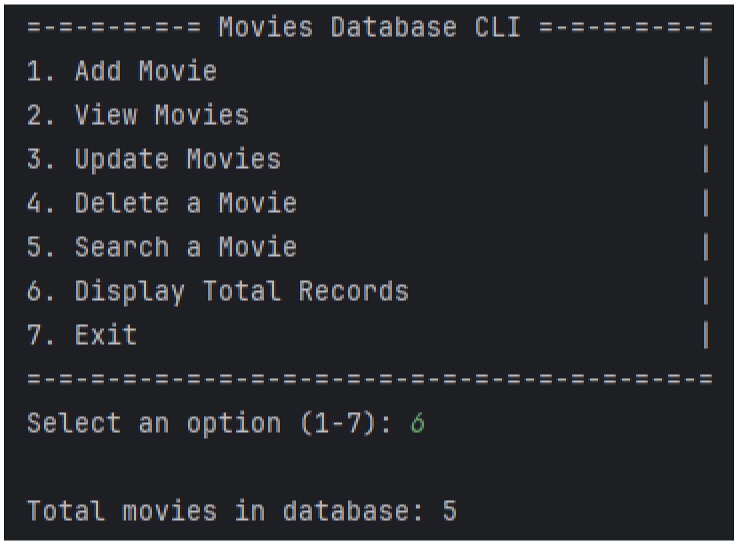
Showing rows 0 - 4 (5 total, Query took 0.0002 seconds.)

```
SELECT * FROM `movies`
```

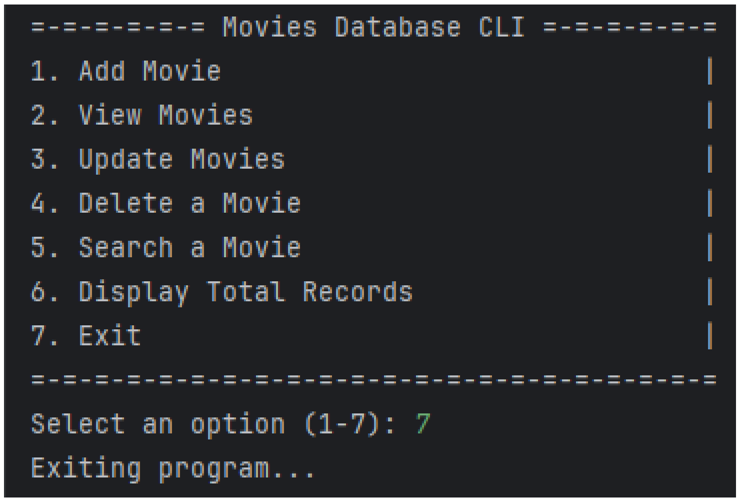
	movie_id	title	main_actor	director	genre	gross_sales	ratings
Edit Copy Delete	3	Twinkling Watermelon	Chung Ah	Shin	Romcom	1000	PG
Edit Copy Delete	4	Goblin	Gong Yu	Shin	Romcom	2000	PG
Edit Copy Delete	5	Pinocchio	Dickie Jones	Mel Blanc	Fantasy	7000	R13
Edit Copy Delete	7	The K2	Ji Chang-wook	Shin	Action	5000	PG
Edit Copy Delete	8	Yongpal	Joo Won	Shin	Thrill	6000	R16



```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                          |
2. View Movies                        |
3. Update Movies                      |
4. Delete a Movie                     |
5. Search a Movie                     |
6. Display Total Records              |
7. Exit                               |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 5

Enter title keyword: Twinkling

--- Search Results ---
(3, 'Twinkling Watermelon', 'Chung Ah', 'Shin', 'Romcom', 1000.0, 'PG')

Total: 1 records
```



```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 6

Total movies in database: 5
```



```
=-=-=-=-=-= Movies Database CLI =-=-=-=-=-=
1. Add Movie                              |
2. View Movies                            |
3. Update Movies                          |
4. Delete a Movie                         |
5. Search a Movie                         |
6. Display Total Records                  |
7. Exit                                   |
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Select an option (1-7): 7
Exiting program...
```