Aquino, Aaron Jan O. December 1,2025

Finals Lab Task 5. CRUD CLI using Python and MySQL

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:

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

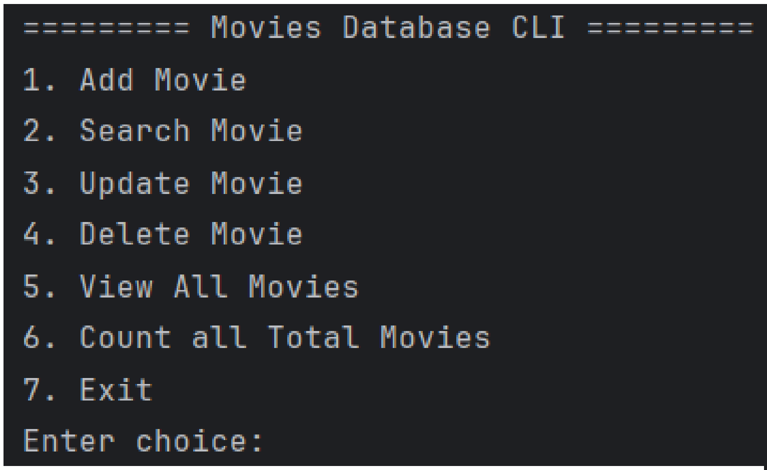
your database – You may use SQL count statement for this.

8. What to submit:

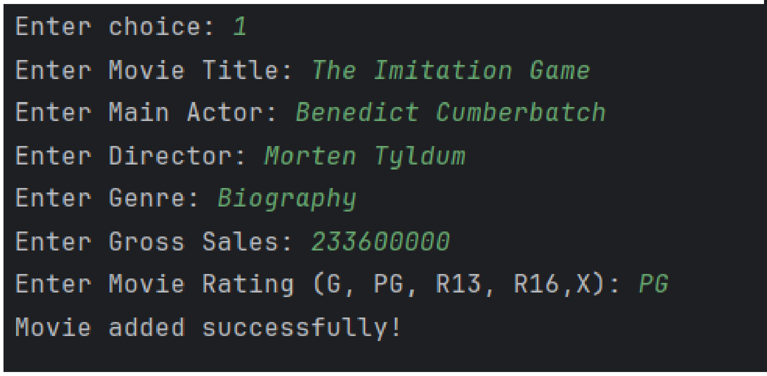
a. UI Menu

b. Sample Output
c. Source Code

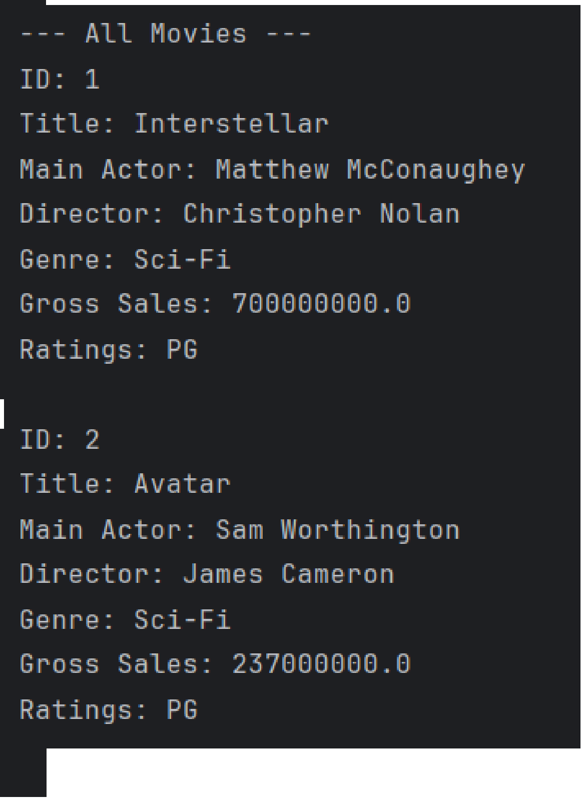
d. Exported sql file



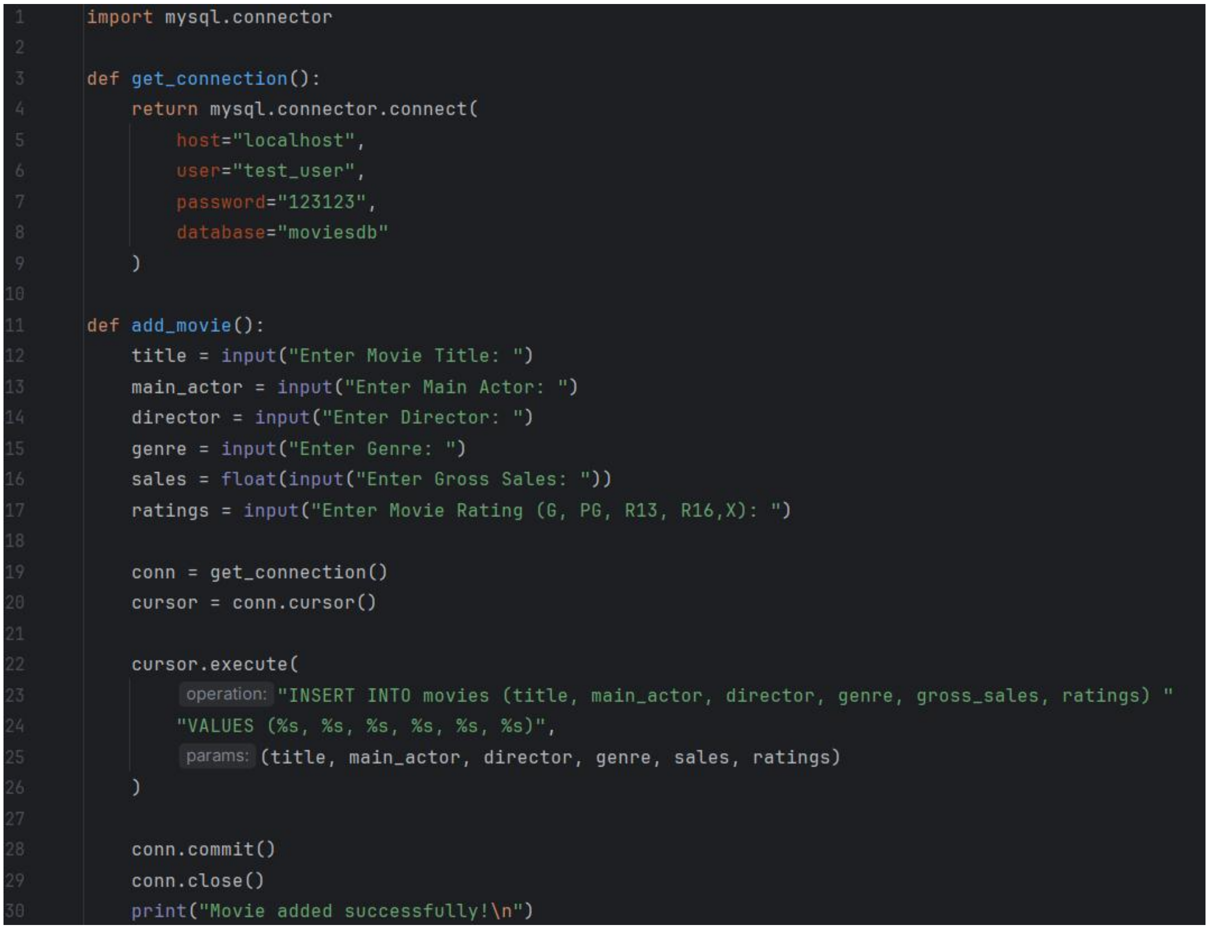
```
========= Movies Database CLI =========
1. Add Movie
2. Search Movie
3. Update Movie
4. Delete Movie
5. View All Movies
6. Count all Total Movies
7. Exit
Enter choice:
```



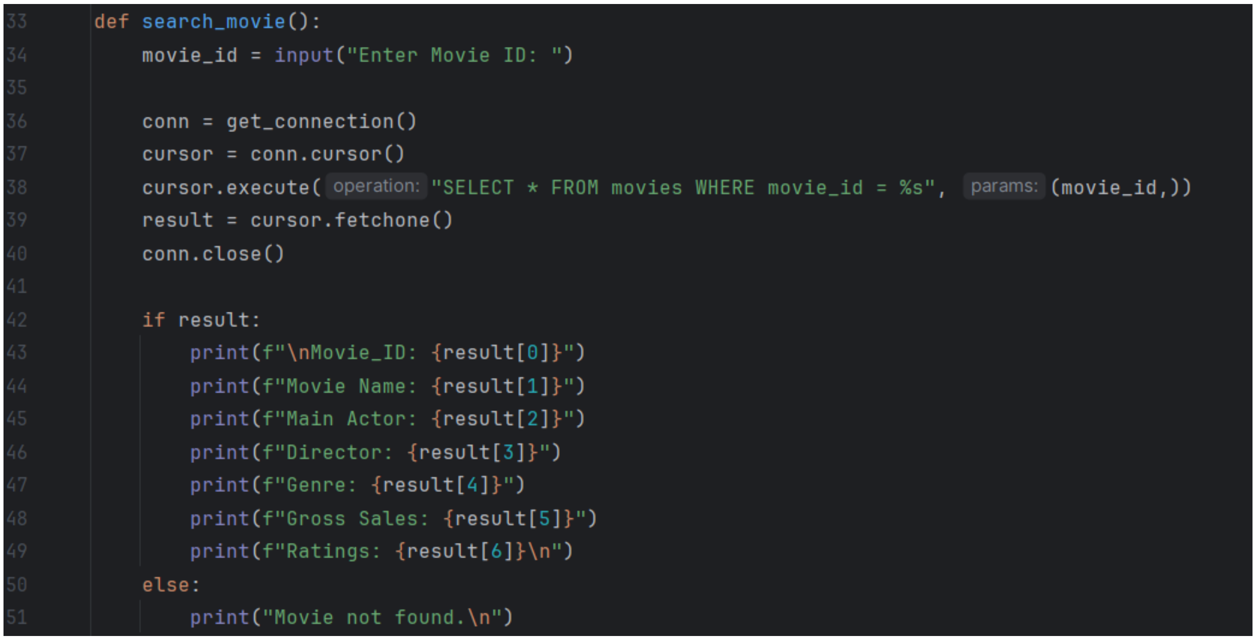
```
Enter choice: 1
Enter Movie Title: The Imitation Game
Enter Main Actor: Benedict Cumberbatch
Enter Director: Morten Tyldum
Enter Genre: Biography
Enter Gross Sales: 233600000
Enter Movie Rating (G, PG, R13, R16,X): PG
Movie added successfully!
```



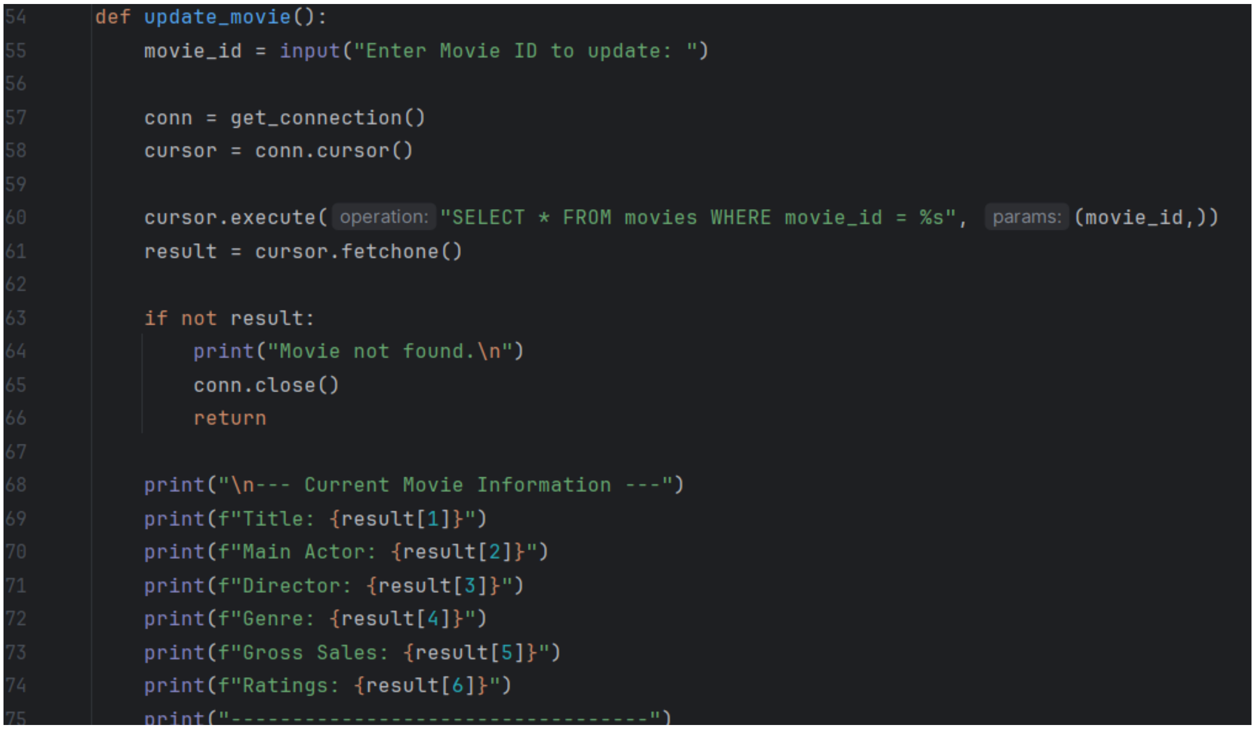
```
--- All Movies ---
ID: 1
Title: Interstellar
Main Actor: Matthew McConaughey
Director: Christopher Nolan
Genre: Sci-Fi
Gross Sales: 700000000.0
Ratings: PG

ID: 2
Title: Avatar
Main Actor: Sam Worthington
Director: James Cameron
Genre: Sci-Fi
Gross Sales: 237000000.0
Ratings: PG
```



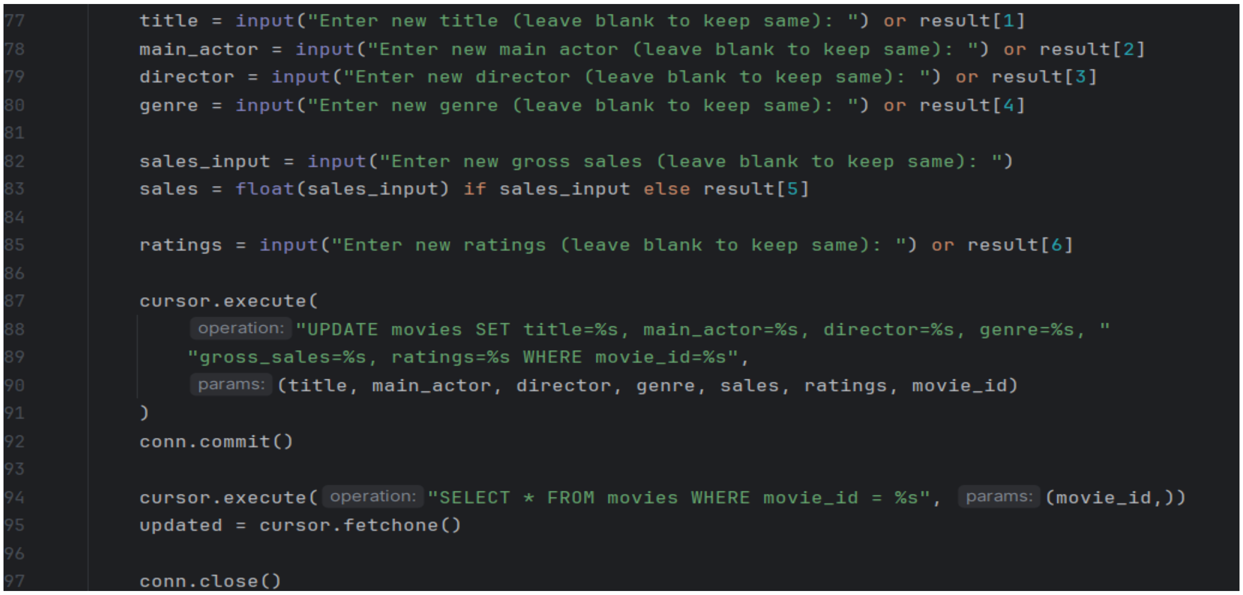
```python
import mysql.connector

def get_connection():
    return mysql.connector.connect(
        host="localhost",
        user="test_user",
        password="123123",
        database="moviesdb"
    )

def add_movie():
    title = input("Enter Movie Title: ")
    main_actor = input("Enter Main Actor: ")
    director = input("Enter Director: ")
    genre = input("Enter Genre: ")
    sales = float(input("Enter Gross Sales: "))
    ratings = input("Enter Movie Rating (G, PG, R13, R16,X): ")

    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute(
        operation: "INSERT INTO movies (title, main_actor, director, genre, gross_sales, ratings) "
        "VALUES (%s, %s, %s, %s, %s, %s)",
        params: (title, main_actor, director, genre, sales, ratings)
    )

    conn.commit()
    conn.close()
    print("Movie added successfully!\n")
```



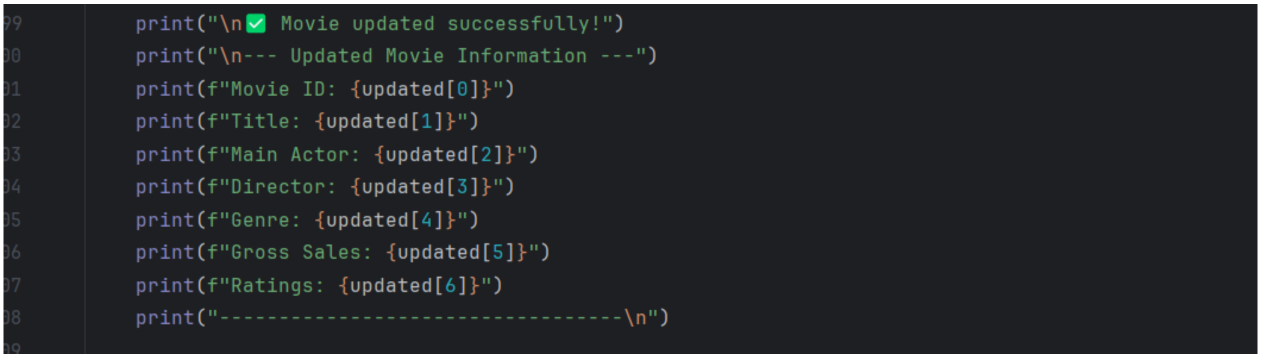
```python
def search_movie():
    movie_id = input("Enter Movie ID: ")

    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute( operation: "SELECT * FROM movies WHERE movie_id = %s",  params: (movie_id,))
    result = cursor.fetchone()
    conn.close()

    if result:
        print(f"\nMovie_ID: {result[0]}")
        print(f"Movie Name: {result[1]}")
        print(f"Main Actor: {result[2]}")
        print(f"Director: {result[3]}")
        print(f"Genre: {result[4]}")
        print(f"Gross Sales: {result[5]}")
        print(f"Ratings: {result[6]}\n")
    else:
        print("Movie not found.\n")
```



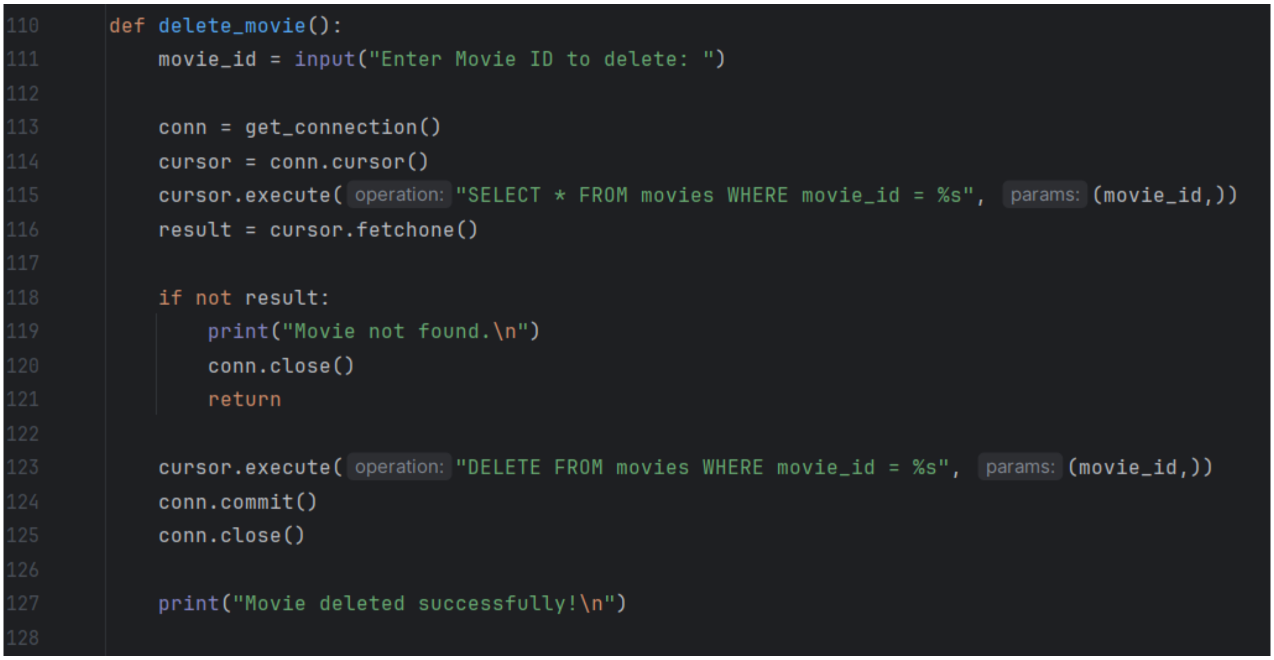
```python
def update_movie():
    movie_id = input("Enter Movie ID to update: ")

    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute( operation: "SELECT * FROM movies WHERE movie_id = %s",  params: (movie_id,))
    result = cursor.fetchone()

    if not result:
        print("Movie not found.\n")
        conn.close()
        return

    print("\n--- Current Movie Information ---")
    print(f"Title: {result[1]}")
    print(f"Main Actor: {result[2]}")
    print(f"Director: {result[3]}")
    print(f"Genre: {result[4]}")
    print(f"Gross Sales: {result[5]}")
    print(f"Ratings: {result[6]}")
    print("-----------------------------------")
```



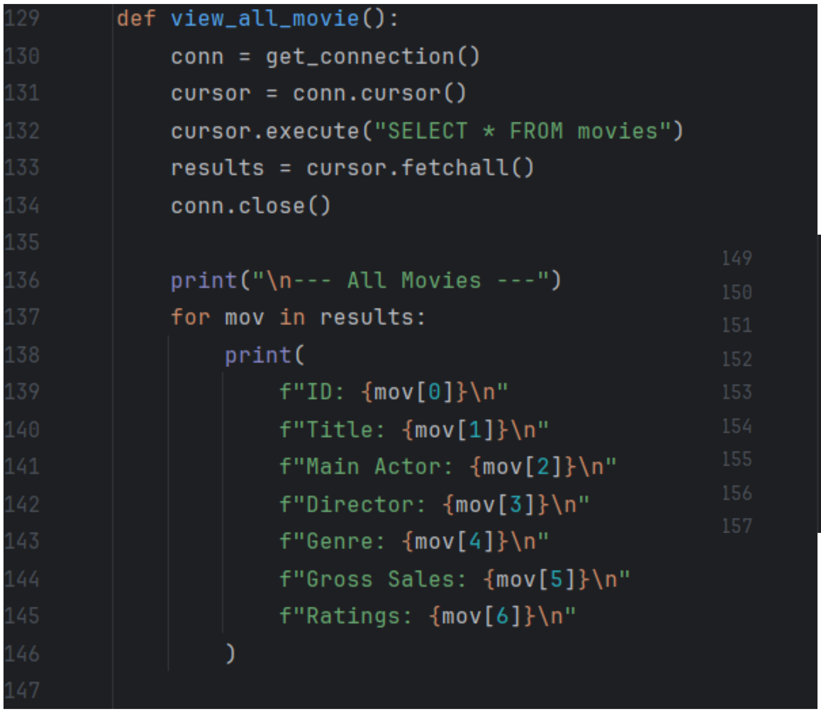
```python
    title = input("Enter new title (leave blank to keep same): ") or result[1]
    main_actor = input("Enter new main actor (leave blank to keep same): ") or result[2]
    director = input("Enter new director (leave blank to keep same): ") or result[3]
    genre = input("Enter new genre (leave blank to keep same): ") or result[4]

    sales_input = input("Enter new gross sales (leave blank to keep same): ")
    sales = float(sales_input) if sales_input else result[5]

    ratings = input("Enter new ratings (leave blank to keep same): ") or result[6]

    cursor.execute(
         operation: "UPDATE movies SET title=%s, main_actor=%s, director=%s, genre=%s, "
        "gross_sales=%s, ratings=%s WHERE movie_id=%s",
         params: (title, main_actor, director, genre, sales, ratings, movie_id)
    )
    conn.commit()

    cursor.execute( operation: "SELECT * FROM movies WHERE movie_id = %s",  params: (movie_id,))
    updated = cursor.fetchone()

    conn.close()
```



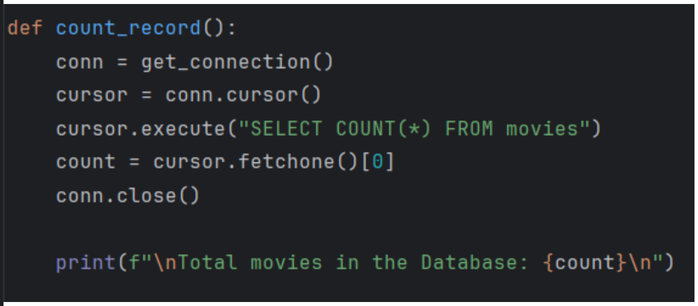
```
    print("\n✅ Movie updated successfully!")
    print("\n--- Updated Movie Information ---")
    print(f"Movie ID: {updated[0]}")
    print(f"Title: {updated[1]}")
    print(f"Main Actor: {updated[2]}")
    print(f"Director: {updated[3]}")
    print(f"Genre: {updated[4]}")
    print(f"Gross Sales: {updated[5]}")
    print(f"Ratings: {updated[6]}")
    print("----------------------------------\n")
```



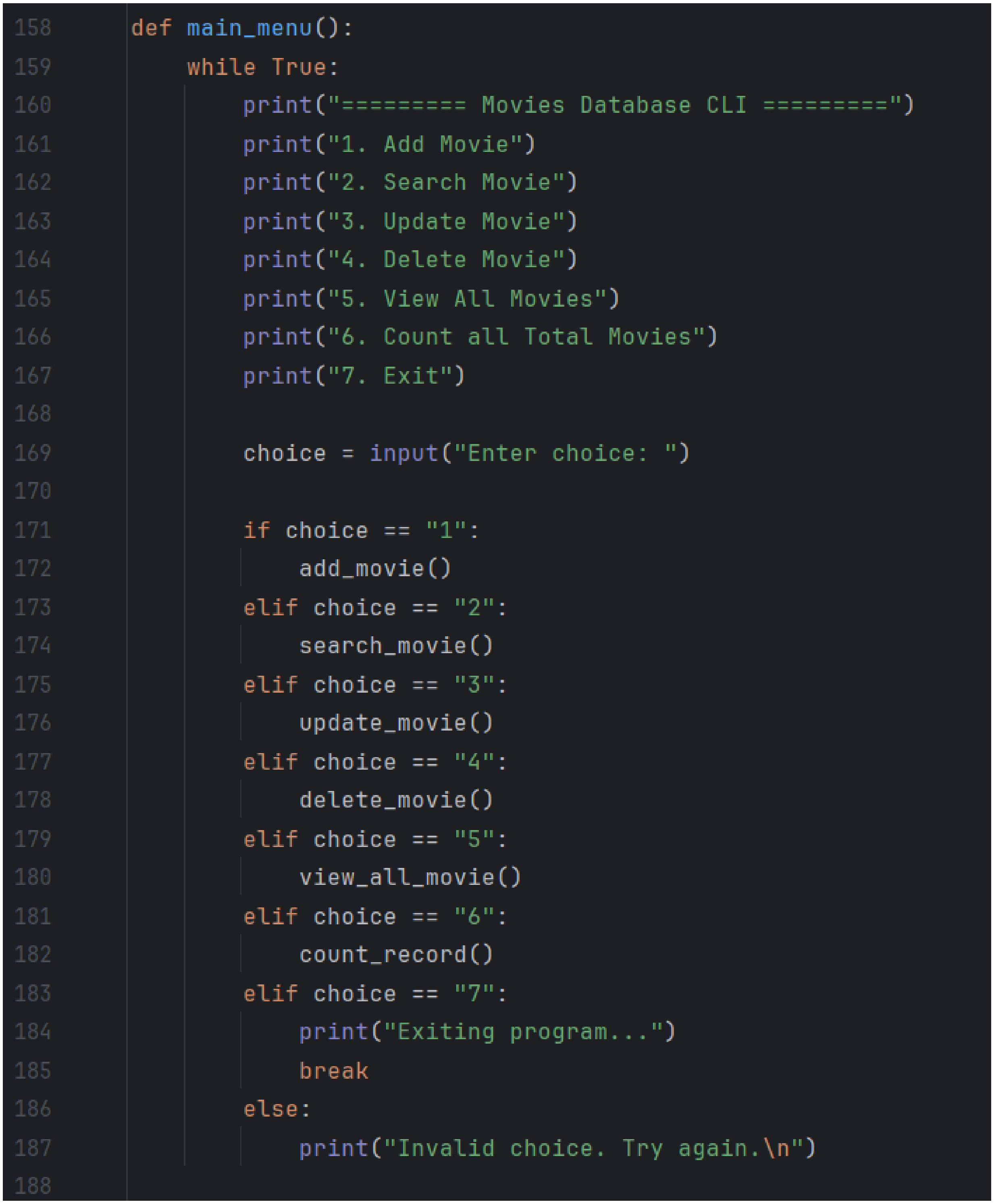
```
def delete_movie():
    movie_id = input("Enter Movie ID to delete: ")

    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute( operation: "SELECT * FROM movies WHERE movie_id = %s",  params: (movie_id,))
    result = cursor.fetchone()

    if not result:
        print("Movie not found.\n")
        conn.close()
        return

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

    print("Movie deleted successfully!\n")
```



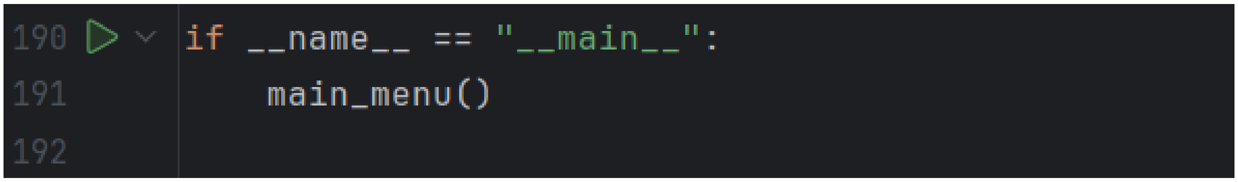
```
def view_all_movie():
    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM movies")
    results = cursor.fetchall()
    conn.close()

    print("\n--- All Movies ---")
    for mov in results:
        print(
            f"ID: {mov[0]}\n"
            f"Title: {mov[1]}\n"
            f"Main Actor: {mov[2]}\n"
            f"Director: {mov[3]}\n"
            f"Genre: {mov[4]}\n"
            f"Gross Sales: {mov[5]}\n"
            f"Ratings: {mov[6]}\n"
        )
```



```
def count_record():
    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute("SELECT COUNT(*) FROM movies")
    count = cursor.fetchone()[0]
    conn.close()

    print(f"\nTotal movies in the Database: {count}\n")
```



```python
def main_menu():
    while True:
        print("========= Movies Database CLI =========")
        print("1. Add Movie")
        print("2. Search Movie")
        print("3. Update Movie")
        print("4. Delete Movie")
        print("5. View All Movies")
        print("6. Count all Total Movies")
        print("7. Exit")

        choice = input("Enter choice: ")

        if choice == "1":
            add_movie()
        elif choice == "2":
            search_movie()
        elif choice == "3":
            update_movie()
        elif choice == "4":
            delete_movie()
        elif choice == "5":
            view_all_movie()
        elif choice == "6":
            count_record()
        elif choice == "7":
            print("Exiting program...")
            break
        else:
            print("Invalid choice. Try again.\n")

```



```python
if __name__ == "__main__":
    main_menu()

```