Johnest Wheyne D. Ogania

BSCS-C204

7OOP

Finals LAB TASK 5: CRUD CLI using Python and MySQL

Problem:

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

Sample Code:

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

6 usages
def create_connection():
    try:
        conn = mysql.connector.connect(
            host="localhost",
            user="test_user",
            password="your_password_here",
            database="moviesDB"
        )
        return conn
    except Error as e:
        print("Error while connecting to MySQL:", e)


1 usage
def add_movie():
    conn = create_connection()
    cursor = conn.cursor()

    print("\n--- ADD NEW MOVIE ---")
    title = input("Enter movie 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): ")

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

    values = (title, actor, director, genre, sales, rating)
    cursor.execute(sql, values)
    conn.commit()
```

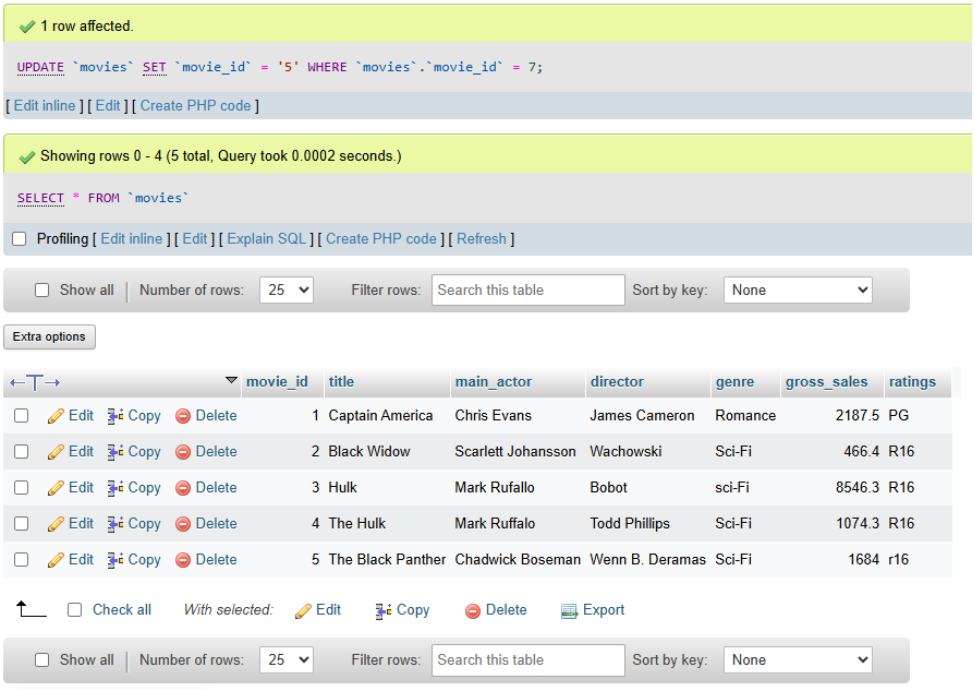
```
    print("Movie added successfully!\n")
    conn.close()


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

    print("\n--- MOVIE LIST ---")
    for row in rows:
        print(f"ID: {row[0]} | Title: {row[1]} | Actor: {row[2]} | Director: {row[3]} | Genre: {row[4]} | Sales: {row[5]} | Rating: {row[6]}")
    print()
    conn.close()


1 usage
def update_movie():
    conn = create_connection()
    cursor = conn.cursor()

    movie_id = input("Enter movie ID to update: ")

    print("\nEnter new details:")
    title = input("New title: ")
    actor = input("New actor: ")
    director = input("New director: ")
    genre = input("New genre: ")
    sales = float(input("New sales: "))
    rating = input("New rating: ")

    sql = """UPDATE movies
             SET title=%s, main_actor=%s, director=%s, genre=%s, gross_sales=%s, ratings=%s
             WHERE movie_id=%s""
```

```
    values = (title, actor, director, genre, sales, rating, movie_id)
    cursor.execute(sql, values)
    conn.commit()

    print("Record updated successfully!\n")
    conn.close()


1 usage
def delete_movie():
    conn = create_connection()
    cursor = conn.cursor()

    movie_id = input("Enter movie ID to delete: ")

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

    print("Record deleted successfully!\n")
    conn.close()


1 usage
def search_movie():
    conn = create_connection()
    cursor = conn.cursor()

    print("\nSearch by: ")
    print("1 - Movie ID")
    print("2 - Title")
    choice = input("Enter choice: ")

    if choice == "1":
        movie_id = input("Enter movie ID: ")
        cursor.execute( operation: "SELECT * FROM movies WHERE movie_id=%s", params: (movie_id,))
    else:
        title = input("Enter title keyword: ")
```

```
        cursor.execute( operation: "SELECT * FROM movies WHERE title LIKE %s",  params: (f"%{title}%",))

    result = cursor.fetchone()

    if result:
        print("\n--- MOVIE FOUND ---")
        print(f"ID: {result[0]} | Title: {result[1]} | Actor: {result[2]} | Director: {result[3]} | Genre: {result[4]} | Sales: {result[5]} | Rating: {result[6]}")
    else:
        print("No matching record found.")

    conn.close()


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

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


1 usage
def main_menu():
    while True:
        print("\nMOVIE DATABASE CRUD APP")
        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 Number of Movies")
        print("7 - Exit")
```

```
        choice = input("Enter your choice: ")

        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("Exiting program...")
            break
        else:
            print("Invalid input. Try again.\n")


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

Sample Output:



1 row affected.

```
UPDATE `movies` SET `movie_id` = '5' WHERE `movies`.`movie_id` = 7;
```

[Edit inline] [Edit] [Create PHP code]

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

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

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

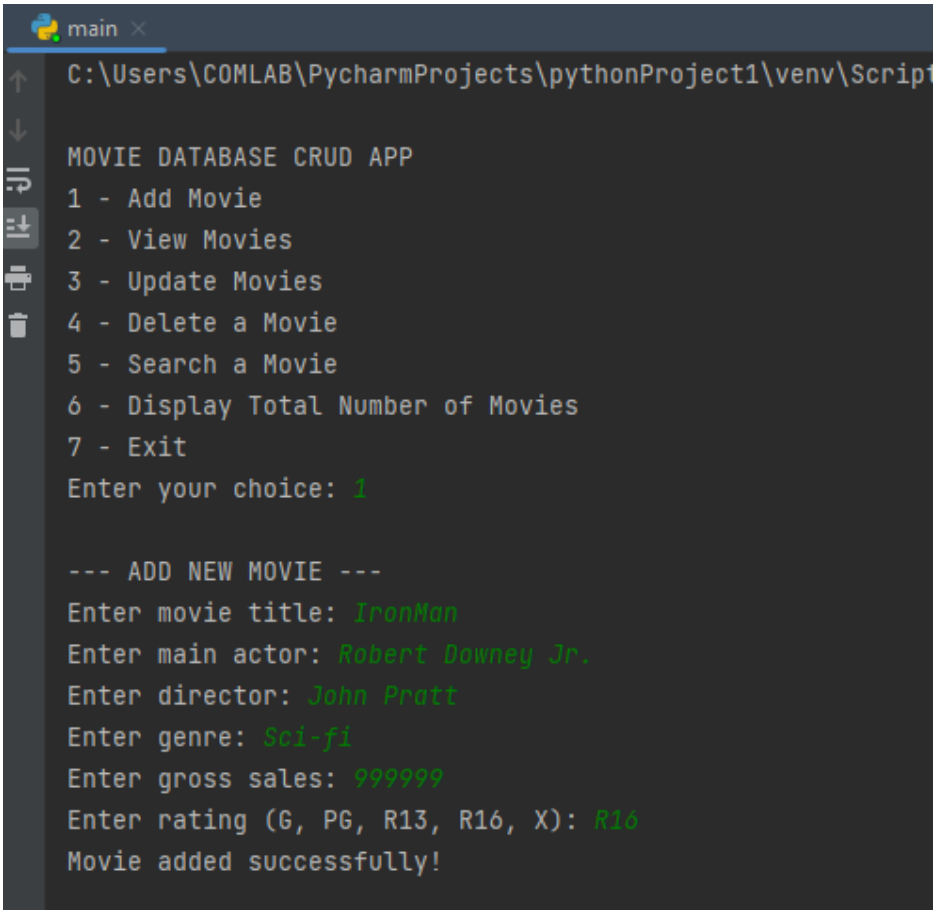
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	movie_id	title	main_actor	director	genre	gross_sales	ratings
Edit Copy Delete	1	Captain America	Chris Evans	James Cameron	Romance	2187.5	PG
Edit Copy Delete	2	Black Widow	Scarlett Johansson	Wachowski	Sci-Fi	466.4	R16
Edit Copy Delete	3	Hulk	Mark Rufallo	Bobot	sci-Fi	8546.3	R16
Edit Copy Delete	4	The Hulk	Mark Ruffalo	Todd Phillips	Sci-Fi	1074.3	R16
Edit Copy Delete	5	The Black Panther	Chadwick Boseman	Wenn B. Deramas	Sci-Fi	1684	r16

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None



```
main
C:\Users\COMLAB\PycharmProjects\pythonProject1\venv\Script

MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 1

--- ADD NEW MOVIE ---
Enter movie title: IronMan
Enter main actor: Robert Downey Jr.
Enter director: John Pratt
Enter genre: Sci-fi
Enter gross sales: 999999
Enter rating (G, PG, R13, R16, X): R16
Movie added successfully!
```

Showing rows 0 - 5 (6 total, Query took 0.0003 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
Edit Copy Delete	1	Captain America	Chris Evans	James Cameron	Romance	2187.5	PG
Edit Copy Delete	2	Black Widow	Scarlett Johansson	Wachowski	Sci-Fi	466.4	R16
Edit Copy Delete	3	Hulk	Mark Rufallo	Bobot	sci-Fi	8546.3	R16
Edit Copy Delete	4	The Hulk	Mark Ruffalo	Todd Phillips	Sci-Fi	1074.3	R16
Edit Copy Delete	5	The Black Panther	Chadwick Boseman	Wenn B. Deramas	Sci-Fi	1684	r16
Edit Copy Delete	8	IronMan	Robert Downey Jr.	John Pratt	Sci-fi	999999	R16

Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 2

--- MOVIE LIST ---
ID: 1 | Title: Captain America | Actor: Chris Evans | Director: James Cameron | Genre: Romance | Sales: 2187.5 | Rating: PG
ID: 2 | Title: Black Widow | Actor: Scarlett Johansson | Director: Wachowski | Genre: Sci-Fi | Sales: 466.4 | Rating: R16
ID: 3 | Title: Hulk | Actor: Mark Rufallo | Director: Bobot | Genre: sci-Fi | Sales: 8546.3 | Rating: R16
ID: 4 | Title: The Hulk | Actor: Mark Ruffalo | Director: Todd Phillips | Genre: Sci-Fi | Sales: 1074.3 | Rating: R16
ID: 5 | Title: The Black Panther | Actor: Chadwick Boseman | Director: Wenn B. Deramas | Genre: Sci-Fi | Sales: 1684.0 | Rating: r16
ID: 8 | Title: IronMan | Actor: Robert Downey Jr. | Director: John Pratt | Genre: Sci-fi | Sales: 999999.0 | Rating: R16
```

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 3
Enter movie ID to update: 4

Enter new details:
New title: Hulk
New actor: John Cena
New director: Patrick
New genre: sci-fi
New sales: 8754.3
New rating: R13
Record updated successfully!
```

Showing rows 0 - 4 (5 total, Query took 0.0003 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
Edit Copy Delete	1	Captain America	Chris Evans	James Cameron	Romance	2187.5	PG
Edit Copy Delete	2	Black Widow	Scarlett Johansson	Wachowski	Sci-Fi	466.4	R16
Edit Copy Delete	4	Hulk	John Cena	Patrick	sci-fi	8754.3	R13
Edit Copy Delete	5	The Black Panther	Chadwick Boseman	Wenn B. Deramas	Sci-Fi	1684	r16
Edit Copy Delete	8	IronMan	Robert Downey Jr.	John Pratt	Sci-fi	999999	R16

Check all *With selected:* Edit Copy Delete Export

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 4
Enter movie ID to delete: 2
Record deleted successfully!
```

Showing rows 0 - 3 (4 total, Query took 0.0003 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
Edit Copy Delete	1	Captain America	Chris Evans	James Cameron	Romance	2187.5	PG
Edit Copy Delete	4	Hulk	John Cena	Patrick	sci-fi	8754.3	R13
Edit Copy Delete	5	The Black Panther	Chadwick Boseman	Wenn B. Deramas	Sci-Fi	1684	r16
Edit Copy Delete	8	IronMan	Robert Downey Jr.	John Pratt	Sci-fi	999999	R16

Check all *With selected:* Edit Copy Delete Export

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 5

Search by:
1 - Movie ID
2 - Title
Enter choice: 2
Enter title keyword: Captain America

--- MOVIE FOUND ---
ID: 1 | Title: Captain America | Actor: Chris Evans | Director: James Cameron | Genre: Romance | Sales: 2187.5 | Rating: PG
```

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 6

Total movies in database: 4
```

```
MOVIE DATABASE CRUD APP
1 - Add Movie
2 - View Movies
3 - Update Movies
4 - Delete a Movie
5 - Search a Movie
6 - Display Total Number of Movies
7 - Exit
Enter your choice: 7
Exiting program...

Process finished with exit code 0
```