

Finals Lab Task 5. CLI using Mysql and Python

Code

```
import mysql.connector

# Function to connect to the MySQL database
def connect_db():
    return mysql.connector.connect(
        host="localhost",
        user="test_user", # MySQL user (replace with your own user if
needed)
        password="123456", # MySQL password (replace with your own password)
        database="moviesDB" # Database name
    )

# Add a new movie record to the database
def add_movie():
    conn = connect_db()
    cursor = conn.cursor()

    title = input("Enter movie title: ")
    main_actor = input("Enter main actor: ")
    director = input("Enter director: ")
    genre = input("Enter genre: ")
    gross_sales = float(input("Enter gross sales: "))
    ratings = input("Enter ratings (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, main_actor, director, genre, gross_sales, ratings))
    conn.commit()
    print("Movie added successfully!\n")
    conn.close()

# View all records in the 'movies' table
def view_movies():
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM movies")
    rows = cursor.fetchall()

    if rows:
        print("\nMovies List:")
        for row in rows:
```

```

        print(row)
else:
    print("\nNo movies found.")
conn.close()

# Update a movie record
def update_movie():
    conn = connect_db()
    cursor = conn.cursor()

    movie_id = int(input("Enter movie ID to update: "))
    title = input("Enter new title: ")
    main_actor = input("Enter new main actor: ")
    director = input("Enter new director: ")
    genre = input("Enter new genre: ")
    gross_sales = float(input("Enter new gross sales: "))
    ratings = input("Enter new ratings (G, PG, R13, R16, X): ")

    cursor.execute("""
        UPDATE movies
        SET title=%s, main_actor=%s, director=%s, genre=%s, gross_sales=%s,
        ratings=%s
        WHERE movie_id=%s
    """, (title, main_actor, director, genre, gross_sales, ratings,
    movie_id))
    conn.commit()
    print("Movie updated successfully!\n")
    conn.close()

# Delete a movie record
def delete_movie():
    conn = connect_db()
    cursor = conn.cursor()

    movie_id = int(input("Enter movie ID to delete: "))
    cursor.execute("DELETE FROM movies WHERE movie_id=%s", (movie_id,))
    conn.commit()
    print("Movie deleted successfully!\n")
    conn.close()

# Search for a movie by title (using LIKE statement)
def search_movie():
    conn = connect_db()
    cursor = conn.cursor()

    search_term = input("Enter movie title to search: ")
    cursor.execute("SELECT * FROM movies WHERE title LIKE %s", ('%' +
    search_term + '%'))
    rows = cursor.fetchall()

    if rows:
        print("\nSearch Results:")
        for row in rows:
            print(row)

```

```

        else:
            print("\nNo matching movies found.")
        conn.close()

# Display the total number of movies in the database
def total_movies():
    conn = connect_db()
    cursor = conn.cursor()

    cursor.execute("SELECT COUNT(*) FROM movies")
    result = cursor.fetchone()
    print(f"\nTotal number of movies: {result[0]}")
    conn.close()

# Main CLI menu for user interaction
def menu():
    while True:
        print("\nMOVIE DATABASE CRUD APP")
        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("Enter 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':
            total_movies()
        elif choice == '7':
            print("Exiting... Goodbye!")
            break
        else:
            print("Invalid choice, please try again.")

# Entry point of the program
if __name__ == "__main__":
    menu()

```

Output:

1.Add New Record

```
MOVIE DATABASE CRUD APP
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
Enter choice: 1
Enter movie title: Spiderboy
Enter main actor: Dexter
Enter director: Christian
Enter genre: Comedy
Enter gross sales: 400000
Enter ratings (G, PG, R13, R16, X): G
Movie added successfully!
```

2. View all records

```
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
Enter choice: 2

Movies List:
(0, 'Spiderboy', 'Dexter', 'Christian', 'Comedy', 400000.0, 'G')
(1, 'Inception', 'Leonardo DiCaprio', 'Christopher Nolan', 'Sci-Fi', 830.0, 'PG-13')
(2, 'The Matrix', 'Keanu Reeves', 'Wachowski Sisters', 'Sci-Fi', 460.0, 'R')
(3, 'Titanic', 'Kate Winslet', 'James Cameron', 'Drama', 2187.0, 'PG-13')
(4, 'The Dark Knight', 'Christian Bale', 'Christopher Nolan', 'Action', 1004.0, 'PG-13')
(5, 'Avengers: Endgame', 'Robert Downey Jr.', 'Anthony Russo', 'Action', 2798.0, 'PG-13')
```

3. Update a Record and show the updates

```
MOVIE DATABASE CRUD APP
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
Enter choice: 3
Enter movie ID to update: 0
Enter new title: Spiderboy
Enter new main actor: Dex
Enter new director: James
Enter new genre: Comedy
Enter new gross sales: 30000
Enter new ratings (G, PG, R13, R16, X): g
Movie updated successfully!
```

4. Search A Record

```
MOVIE DATABASE CRUD APP
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
Enter choice: 5
Enter movie title to search: Spiderboy

Search Results:
(0, 'Spiderboy', 'Dex', 'James', 'Comedy', 30000.0, 'g')
```

5. Display Total Numbers of Movies stored in the database

```
MOVIE DATABASE CRUD APP
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
Enter choice: 6

Total number of movies: 6
```