Cabenta, Joseph S.
BSCS – C204

December 1, 2025
7OOP

FINALS LAB TASK 5

I. 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

II. CODE

movieCli.py × moviesdb.sql

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

# ----- CONFIG DEFAULTS -----
DB_CONFIG = {
    'host': 'localhost',
    'user': 'test_user',
    'password': 'StrongPassword123',
    'database': 'moviesDB',
    'port': 3306
}
# ---------------------------------------------------------------

def ask_db_credentials():  1 usage
    """Ask user for DB credentials (allow ENTER to keep defaults)."""

    print("Enter database connection details (press Enter to keep default shown in brackets).")

    # Host
    host = input(f"Host [{DB_CONFIG['host']}]: ").strip() or DB_CONFIG['host']

    # User
    user = input(f"User [{DB_CONFIG['user']}]: ").strip() or DB_CONFIG['user']

    # Password (plain input for PyCharm compatibility)
    pwd_default = DB_CONFIG.get('password') or ''
    print("(Leave Password blank to use the default password.)")
    password = input("Password: ").strip()
    if password == "" and pwd_default != "":
        password = pwd_default

    # Database
    database = input(f"Database [{DB_CONFIG['database']}]: ").strip() or DB_CONFIG['database']
    # Port
    port_input = input(f"Port [{DB_CONFIG['port']}]: ").strip()
    port = int(port_input) if port_input else DB_CONFIG['port']

    return {
        'host': host,
        'user': user,
        'password': password,
        'database': database,
        'port': port
    }


def connect_db(cfg):  1 usage
    try:
        conn = mysql.connector.connect(**cfg)
        return conn
    except Error as e:
        print("ERROR: Could not connect to database:", e)
        return None

def run_query(conn, query, params=None, fetch=False):  10 usages
    try:
        cur = conn.cursor(dictionary=True)
        cur.execute(query, params or ())
        if fetch:
            rows = cur.fetchall()
            cur.close()
            return rows
        else:
            conn.commit()
            affected = cur.rowcount
            cur.close()
            return affected
```

```
    except Error as e:
        print("Database error:", e)
        return None

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

def input_movie_fields(existing=None):  2 usages
    # existing: dict used for default values on update
    def get(field_name, default=""):
        prompt = f"{field_name}"
        if default not in ("", None):
            prompt += f" [{default}]"
        prompt += ": "
        val = input(prompt).strip()
        return val if val != "" else default

    title = get( field_name: "Title", existing.get('title') if existing else "")
    main_actor = get( field_name: "Main Actor", existing.get('main_actor') if existing else "")
    director = get( field_name: "Director", existing.get('director') if existing else "")
    genre = get( field_name: "Genre", existing.get('genre') if existing else "")
    gross_sales_raw = get( field_name: "Gross Sales (numeric)", existing.get('gross_sales') if existing else "")
    ratings = get( field_name: "Ratings (G, PG, R, PG-13)", existing.get('ratings') if existing else "")

    gross_sales = None
```

```
    if gross_sales_raw != "" and gross_sales_raw is not None:
        try:
            gross_sales = float(gross_sales_raw)
        except ValueError:
            print("Warning: Gross Sales invalid, stored as NULL.")
            gross_sales = None

    return {
        'title': title,
        'main_actor': main_actor,
        'director': director,
        'genre': genre,
        'gross_sales': gross_sales,
        'ratings': ratings
    }

def add_movie(conn):  1 usage
    print("\n-- Add Movie --")
    data = input_movie_fields()
    if not data['title'] or not data['main_actor'] or not data['director'] or not data['genre']:
        print("Title, Main Actor, Director and Genre are required.")
        return
    q = """INSERT INTO movies (title, main_actor, director, genre, gross_sales, ratings)
           VALUES (%s,%s,%s,%s,%s,%s)"""
    res = run_query(conn, q,  params: (data['title'], data['main_actor'], data['director'],
                                       data['genre'], data['gross_sales'], data['ratings']))
    if res is not None:
        print("Movie added successfully (rows affected):", res)

def view_movies(conn):  1 usage
    print("\n-- All Movies --")
    rows = run_query(conn,  query: "SELECT * FROM movies ORDER BY movie_id", fetch=True)
    if rows is None:
        return
```

```
        if not rows:
            print("No records.")
            return
        # print header
        print("{:<6} {:<30} {:<20} {:<20} {:<12} {:>10} {:<8}".format(
            *args: "ID", "Title", "Main Actor", "Director", "Genre", "GrossSales", "Rate"))
        print("-"*110)
        for r in rows:
            print("{:<6} {:<30} {:<20} {:<20} {:<12} {:>10} {:<8}".format(
                *args: r['movie_id'],
                (r['title'] or "")[:30],
                (r['main_actor'] or "")[:20],
                (r['director'] or "")[:20],
                (r['genre'] or "")[:12],
                f"{r['gross_sales']:.2f}" if r['gross_sales'] is not None else "",
                r['ratings'] or ""
            ))

    def update_movie(conn):  1 usage
        print("\n-- Update Movie --")
        try:
            mid = int(input("Enter movie_id to update: ").strip())
        except ValueError:
            print("Invalid id.")
            return
        rows = run_query(conn, query: "SELECT * FROM movies WHERE movie_id=%s", params: (mid,), fetch=True)
        if rows is None:
            return
        if len(rows) == 0:
            print("No movie with that id.")
            return
        existing = rows[0]
        print("Current values (press Enter to keep current):")
        new_data = input_movie_fields(existing)
```

```
        if not new_data['title'] or not new_data['main_actor'] or not new_data['director'] or not new_data['genre']:
            print("Title, Main Actor, Director and Genre are required.")
            return
        q = """UPDATE movies SET title=%s, main_actor=%s, director=%s, genre=%s,
                gross_sales=%s, ratings=%s WHERE movie_id=%s"""
        res = run_query(conn, q, params: (new_data['title'], new_data['main_actor'], new_data['director'],
                                  new_data['genre'], new_data['gross_sales'], new_data['ratings'], mid))
        if res is not None:
            print("Movie updated (rows affected):", res)

    def delete_movie(conn):  1 usage
        print("\n-- Delete Movie --")
        try:
            mid = int(input("Enter movie_id to delete: ").strip())
        except ValueError:
            print("Invalid id.")
            return
        rows = run_query(conn, query: "SELECT title FROM movies WHERE movie_id=%s", params: (mid,), fetch=True)
        if rows is None:
            return
        if not rows:
            print("No movie with that id.")
            return
        confirm = input(f"Are you sure you want to delete '{rows[0]['title']}'? (y/N): ").strip().lower()
        if confirm != 'y':
            print("Deletion cancelled.")
            return
        res = run_query(conn, query: "DELETE FROM movies WHERE movie_id=%s", params: (mid,))
        if res is not None:
            print("Deleted (rows affected):", res)

    def search_movie(conn):  1 usage
        print("\n-- Search Movie --")
        choice = input("Search by 'id' or enter text to search title/actor/director: ").strip()
```

```
    if choice == "":
        print("No input.")
        return

    # If the user entered only digits, treat it as an ID search (convenience).
    if choice.isdigit():
        movie_id = int(choice)
        rows = run_query(conn, query: "SELECT * FROM movies WHERE movie_id=%s", params: (movie_id,), fetch=True)

    # If user literally typed 'id', ask for the id next.
    elif choice.lower() == 'id':
        try:
            movie_id = int(input("Enter movie_id: ").strip())
        except ValueError:
            print("Invalid id.")
            return
        rows = run_query(conn, query: "SELECT * FROM movies WHERE movie_id=%s", params: (movie_id,), fetch=True)

    # Otherwise treat the input as a text search (partial match)
    else:
        term = "%" + choice + "%"
        rows = run_query(conn,
                         query: "SELECT * FROM movies WHERE title LIKE %s OR main_actor LIKE %s OR director LIKE %s",
                         params: (term, term, term), fetch=True)

    if rows is None:
        return
    if not rows:
        print("No records found.")
        return

    # Print results
    print("{:<6} {:<30} {:<20} {:<20} {:<12} {:>10} {:<8}".format(
        *args: "ID", "Title", "Main Actor", "Director", "Genre", "GrossSales", "Rate"))
```

```
    print("-" * 110)
    for r in rows:
        print("{:<6} {:<30} {:<20} {:<20} {:<12} {:>10} {:<8}".format(
            *args: r['movie_id'],
            (r['title'] or "")[:30],
            (r['main_actor'] or "")[:20],
            (r['director'] or "")[:20],
            (r['genre'] or "")[:12],
            f"{r['gross_sales']:.2f}" if r['gross_sales'] is not None else "",
            r['ratings'] or ""
        ))

def display_count(conn):  1 usage
    rows = run_query(conn, query: "SELECT COUNT(*) AS total FROM movies", fetch=True)
    if rows is None:
        return
    print("Total number of movies:", rows[0]['total'])

def main():  1 usage
    cfg = ask_db_credentials()
    conn = connect_db(cfg)
    if conn is None:
        print("Exiting.")
        sys.exit(1)

    print("Connected to database.")
    try:
        while True:
            choice = menu()
            if choice == '1':
                add_movie(conn)
            elif choice == '2':
                view_movies(conn)
            elif choice == '3':
```

```
                update_movie(conn)
            elif choice == '4':
                delete_movie(conn)
            elif choice == '5':
                search_movie(conn)
            elif choice == '6':
                display_count(conn)
            elif choice == '7':
                print("Goodbye.")
                break
            else:
                print("Invalid selection. Choose 1-7.")
        finally:
            conn.close()

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

III. SAMPLE OUTPUT

```
"D:\School\School Works\Pycharm\.venv\Scripts\python.exe" "D:\School\School Works\Pycharm\Finals\CLI Act\movieCL
Enter database connection details (press Enter to keep default shown in brackets).
Host [localhost]:
User [test_user]:
(Leave Password blank to use the default password.)
Password:
Database [moviesDB]:
Port [3306]:
Connected to database.

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

-- All Movies --
ID   Title                Main Actor          Director            Genre       GrossSales Rate
-------------------------------------------------------------------------------------------
1    Blade Runner         Harrison Ford       Ridley Scott        Sci-Fi          327.50 R
2    The Matrix           Keanu Reeves        Wachowski           Sci-Fi          463.50 R
3    Spirited Away        Rumi Hiiragi        Hayao Miyazaki      Animation       347.70 G
4    The Godfather        Marlon Brando       Francis Ford Coppola Crime          246.00 R
5    Interstellar         Matthew McConaughey Christopher Nolan   Sci-Fi          677.50 PG-13

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

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

-- Add Movie --
Title: The Lost Kingdom
Main Actor: Daniel Reyes
Director: Maria Santos
Genre: Adventure
Gross Sales (numeric): 152.7
Ratings (G, PG, R, PG-13): PG
Movie added successfully (rows affected): 1

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

-- All Movies --
ID     Title                     Main Actor          Director            Genre       GrossSales Rate
-------------------------------------------------------------------------------------------------------
1      Blade Runner              Harrison Ford       Ridley Scott        Sci-Fi          327.50 R
2      The Matrix                Keanu Reeves        Wachowski           Sci-Fi          463.50 R
3      Spirited Away             Rumi Hiiragi        Hayao Miyazaki      Animation       347.70 G
4      The Godfather             Marlon Brando       Francis Ford Coppola Crime          246.00 R
5      Interstellar              Matthew McConaughey Christopher Nolan   Sci-Fi          677.50 PG-13
7      The Lost Kingdom          Daniel Reyes        Maria Santos        Adventure       152.70 PG

----- MOVIE DATABASE CLI -----
1. Add Movie
2. View All Movies
3. Update Movie
```

```
4. Delete Movie
5. Search Movie
6. Display Total Records
7. Exit
Select an option (1-7): 3

-- Update Movie --
Enter movie_id to update: 7
Current values (press Enter to keep current):
Title [The Lost Kingdom]:
Main Actor [Daniel Reyes]:
Director [Maria Santos]:
Genre [Adventure]: Drama
Gross Sales (numeric) [152.7]: 160.9
Ratings (G, PG, R, PG-13) [PG]: PG-13
Movie updated (rows affected): 1

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

-- All Movies --
ID     Title                     Main Actor          Director            Genre       GrossSales Rate
-------------------------------------------------------------------------------------------------------
1      Blade Runner              Harrison Ford       Ridley Scott        Sci-Fi          327.50 R
2      The Matrix                Keanu Reeves        Wachowski           Sci-Fi          463.50 R
3      Spirited Away             Rumi Hiiragi        Hayao Miyazaki      Animation       347.70 G
4      The Godfather             Marlon Brando       Francis Ford Coppola Crime          246.00 R
5      Interstellar              Matthew McConaughey Christopher Nolan   Sci-Fi          677.50 PG-13
7      The Lost Kingdom          Daniel Reyes        Maria Santos        Drama           160.90 PG-13
```

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

-- Delete Movie --
Enter movie_id to delete: 7
Are you sure you want to delete 'The Lost Kingdom'? (y/N): Y
\Deleted (rows affected): 1

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

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

```
-- All Movies --
ID   Title                Main Actor          Director            Genre       GrossSales Rate
-------------------------------------------------------------------------------------------
1    Blade Runner         Harrison Ford       Ridley Scott        Sci-Fi          327.50 R
2    The Matrix           Keanu Reeves        Wachowski           Sci-Fi          463.50 R
3    Spirited Away        Rumi Hiiragi        Hayao Miyazaki      Animation       347.70 G
4    The Godfather        Marlon Brando       Francis Ford Coppola Crime          246.00 R
5    Interstellar         Matthew McConaughey Christopher Nolan   Sci-Fi          677.50 PG-13

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

-- Search Movie --
Search by 'id' or enter text to search title/actor/director: 4
ID   Title                Main Actor          Director            Genre       GrossSales Rate
-------------------------------------------------------------------------------------------
4    The Godfather        Marlon Brando       Francis Ford Coppola Crime          246.00 R

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

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

Process finished with exit code 0
```