Finals Lab Task 6. Connection Mysql using Python using CLI

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
    d. Make sure you have a database in Mysql named testDB with an employees table with at-least 10 records (You may import the attached SQL file to start with

2. Guided by the Demo code attached in this task. Test_DemoDB.py
3. Kindly continue working on the code that will allow the user to navigate through the Database and perform CRUD operations. Follow the following **CLI Menu Options:**
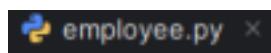
```
----- Employee Manager -----
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Exit
Select an option (1-5):
```

4. The user should be able to add New Records when 1 is selected and show the records, 2- Display all records, 3 Update a Record and show the updates, 4 – Delete a record 5- Exit
5. For additional challenge, Task – You are tasked 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.

PROGRAM CODE:

🐍 employee.py ×

```python
import mysql.connector

conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="testdb"
)

cursor = conn.cursor()

def add_employee():  1 usage
    name = input("Enter name: ")
    emp_id = input("Enter employee ID: ")
    salary = float(input("Enter salary: "))

    cursor.execute(
        operation: "INSERT INTO employees (name, emp_id, salary) VALUES (%s, %s, %s)",
        params: (name, emp_id, salary)
    )
    conn.commit()
    print("\n✓ Employee added successfully!\n")

def view_employees():  1 usage
    cursor.execute("SELECT * FROM employees")
    rows = cursor.fetchall()

    print("\n=== EMPLOYEES LIST ===")
    if rows:
        for row in rows:
            print(row)
    else:
        print("No employees found.")
    print()

def update_employee():  1 usage
    emp_id = input("Enter employee ID to update: ")

    cursor.execute( operation: "SELECT * FROM employees WHERE emp_id=%s",  params: (emp_id,))
    result = cursor.fetchone()
```

```python
        if not result:
            print("\n⚠ Employee not found!\n")
            return

        print("\nCurrent Info:", result)

        new_name = input("Enter new name: ")
        new_salary = float(input("Enter new salary: "))

        cursor.execute(
            operation: "UPDATE employees SET name=%s, salary=%s WHERE emp_id=%s",
            params: (new_name, new_salary, emp_id)
        )
        conn.commit()
        print("\n✓ Employee updated successfully!\n")

def delete_employee():  1 usage
    emp_id = input("Enter employee ID to delete: ")

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

    if not result:
        print("\n⚠ Employee not found!\n")
        return

    cursor.execute( operation: "DELETE FROM employees WHERE emp_id=%s", params: (emp_id,))
    conn.commit()
    print("\n✓ Employee deleted successfully!\n")

def search_employee():  1 usage
    print("\nSearch by:")
    print("1 - Name")
    print("2 - Employee ID")
    choice = input("Choose search type: ")
```

```python
        if choice == "1":
            name = input("Enter name to search: ")
            cursor.execute( operation: "SELECT * FROM employees WHERE name LIKE %s",  params: (f"%{name}%",))
        elif choice == "2":
            emp_id = input("Enter employee ID: ")
            cursor.execute( operation: "SELECT * FROM employees WHERE emp_id=%s",  params: (emp_id,))
        else:
            print("Invalid option!\n")
            return

        results = cursor.fetchall()

        print("\n=== SEARCH RESULT ===")
        if results:
            for r in results:
                print(r)
        else:
            print("No matching records found.")
        print()

def main():  1 usage
    while True:
        print("\n====== EMPLOYEE DATABASE MENU ======")
        print("1 - Add New Employee")
        print("2 - Display All Employees")
        print("3 - Update Employee")
        print("4 - Delete Employee")
        print("5 - Search Employee")
        print("6 - Exit")

        option = input("Choose an option: ")

        if option == "1":
            add_employee()
        elif option == "2":
            view_employees()
        elif option == "3":
            update_employee()
        elif option == "4":
            delete_employee()
        elif option == "5":
            search_employee()
        elif option == "6":
            print("\nExiting program...")
            break
        else:
            print("\nInvalid option. Try again.\n")

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

SAMPLE OUTPUTs:

testrun1(Display):

```
====== EMPLOYEE DATABASE MENU ======
1 - Add New Employee
2 - Display All Employees
3 - Update Employee
4 - Delete Employee
5 - Search Employee
6 - Exit
Choose an option: 2

=== EMPLOYEES LIST ===
('Bradd Pitt', 101, 3000.0)
('Tom Cruise', 102, 15000.0)
('Patchak', 103, 3000.0)
('Angeee', 107, 5000.0)
('Trish', 108, 600.0)
```

```
====== EMPLOYEE DATABASE MENU ======
 - Add New Employee
 - Display All Employees
 - Update Employee
 - Delete Employee
 - Search Employee
 - Exit
Choose an option: 5

Search by:
 - Name
 - Employee ID
Choose search type: 1
Enter name to search: Brad Pitt

=== SEARCH RESULT ===
No matching records found.
```

testrun2(Search):

testrun3(Add):

```
====== EMPLOYEE DATABASE MENU ======
1 - Add New Employee
2 - Display All Employees
3 - Update Employee
4 - Delete Employee
5 - Search Employee
6 - Exit
Choose an option: 1
Enter name: Rice Shai
Enter employee ID: 234
Enter salary: 1900

✓ Employee added successfully!



====== EMPLOYEE DATABASE MENU ======
1 - Add New Employee
2 - Display All Employees
3 - Update Employee
4 - Delete Employee
5 - Search Employee
6 - Exit
Choose an option: 1
Enter name: Sak Chi
Enter employee ID: 567
Enter salary: 5009

✓ Employee added successfully!
```

(Evidence)

				name	emp_id	salary
☐	Edit	Copy	Delete	Bradd Pitt	101	3000
☐	Edit	Copy	Delete	Tom Cruise	102	15000
☐	Edit	Copy	Delete	Patchak	103	3000
☐	Edit	Copy	Delete	Angeee	107	5000
☐	Edit	Copy	Delete	Trish	108	600
☐	Edit	Copy	Delete	Rice Shai	234	1900
☐	Edit	Copy	Delete	Sak Chi	567	5009

testrun4(Update):

```
====== EMPLOYEE DATABASE MENU ======
1 - Add New Employee
2 - Display All Employees
3 - Update Employee
4 - Delete Employee
5 - Search Employee
6 - Exit
Choose an option: 3
Enter employee ID to update: 567

Current Info: ('Sak Chi', 567, 5009.0)
Enter new name: Sakura Chi
Enter new salary: 5000

✓ Employee updated successfully!
```

(Evidence)

				name	emp_id	salary
☐	Edit	Copy	Delete	Bradd Pitt	101	3000
☐	Edit	Copy	Delete	Tom Cruise	102	15000
☐	Edit	Copy	Delete	Patchak	103	3000
☐	Edit	Copy	Delete	Angeee	107	5000
☐	Edit	Copy	Delete	Trish	108	600
☐	Edit	Copy	Delete	Rice Shai	234	1900
☐	Edit	Copy	Delete	Sakura Chi	567	5000

testrun5(Delete):

```
====== EMPLOYEE DATABASE MENU ======
1 - Add New Employee
2 - Display All Employees
3 - Update Employee
4 - Delete Employee
5 - Search Employee
6 - Exit
Choose an option: 4
Enter employee ID to delete: 234

✓ Employee deleted successfully!
```

(Evidence)

←T→				name	emp_id	salary
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Bradd Pitt	101	3000
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Tom Cruise	102	15000
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Patchak	103	3000
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Angeee	107	5000
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Trish	108	600
☐	🖉 Edit	⌷ᶜ Copy	⊖ Delete	Sakura Chi	567	5000