Sukkur Institute of Business Administration University
Department of Computer Science
BS – II (CS/SE/AI) Spring 2024
Object Oriented Programming
Lab # 12: To become familiar with JDBC (Java DataBase Connectivity)
Instructor: Engr. Zainab Umair Kamangar

Lab Report Rubrics (Add the points in each column, then add across the bottom row to find the total score)					**Total Marks**
S.No	**Criterion**	**0.5**	**0.25**	**0.125**	
1	Accuracy	☐ Desired output	☐ Minor mistakes	☐ Critical mistakes	
2	Timing	☐ Submitted within the given time	☐ 1 day late	☐ More than 1 day late	

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Objectives

After performing this lab, students will be able to understand,

- MySQL and Connectors
- Overview of JDBC
- JDBC with GUI

Pre Requisite

Before we actually start JDBC, You must have these things installed on your PC.

1. DBMS (In our case, MySQL)

https://dev.mysql.com/downloads/installer/   (Installer link)
https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html (Installation Steps)

For your convenience, I have shared a recording that shows the steps to install MYSQL on your machines. And getting started with first jdbc project.

https://www.youtube.com/playlist?list=PLRH4SPkZ-XBoL5xULo70igLS96O7CFcAV

2. Java connectors (JDBC Drivers)

https://dev.mysql.com/downloads/connector/j/
Open this link, and choose platform independent from dropdown menu and download the pointed file.
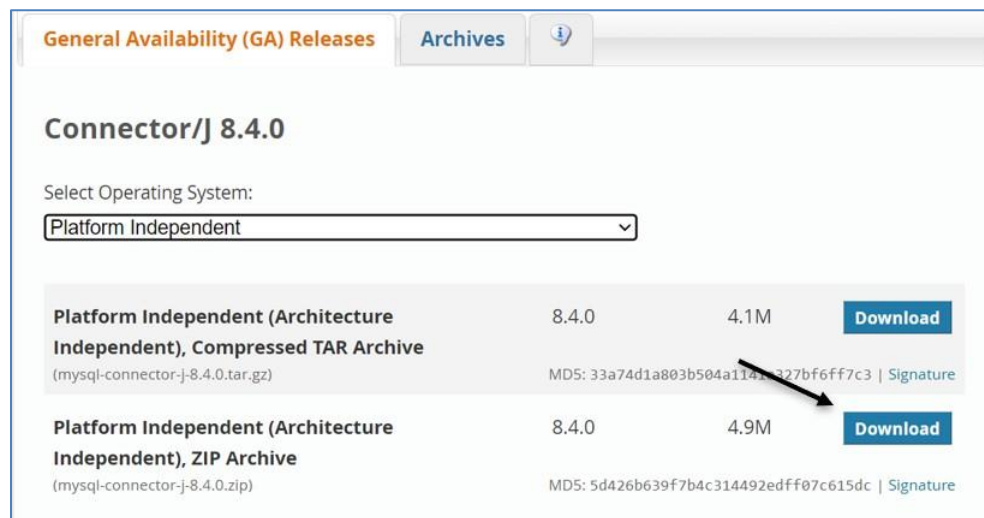
At the end, you will get jar files for drivers. To connect java application with the mysql database, **mysqlconnector.jar** file is required to be loaded.

Overview of JDBC

To connect Java application with the MySQL database, we need to follow 5 following steps.

In this example we are using MySql as the database. So we need to know following informations for the mysql database:

1. **Driver class:** The driver class for the mysql database is "**com.mysql.cj.jdbc.Driver**"

2. **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/lab** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.

3. **Username:** The default username for the mysql database is **root**.

4. **Password:** It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

Let's first create a table in the mysql database, but before creating table, we need to create database first. For that you may open MYSQL client and follow the steps.

```
create database lab;
use lab;
create table emp(id int(10),name varchar(40),age int(3));
```

Example

MySQL 8.0 Command Line Client

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database Animals
    -> ;
Query OK, 1 row affected (0.02 sec)

mysql> create table Pet (id int, pet_name varchar(30), sound varchar(50));
ERROR 1046 (3D000): No database selected
mysql> use database Animals
ERROR 1049 (42000): Unknown database 'database'
mysql> use Animals
Database changed
```

Example to Connect Java Application with mysql database
In this example, lab is the database name, root is the username and password both.

```java
import java.sql.*;
class MysqlCon{
    public static void main(String args[]){
        try{
          Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con=DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/lab","root","");
            //here lab is database name, root is username and password
        is empty (if any password set you can use it)
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("select * from emp");
            while(rs.next())
            System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs
            .getString(3));
            con.close();
            }
        catch (ClassNotFoundException e){
                    System.out.println(e.getMessage());
                }
                catch (Exception e) {
                        System.out.println(e.getMessage());
                }
    }
}
```

The above example will fetch all the records of emp table.

To connect java application with the mysql database, **mysqlconnector.jar** file is required to be
loaded. For that google mysql-connector-java-8.0.25 and download it
If you are not using any IDE then you need to follow following steps:

1. Download mysql-connector.jar file.
2. Move it(jar) to your project directory where main class exist.
3. Run command **javac MysqlCon.java**
4. Run command **java -cp ".;mysql-connector.jar" MysqlCon**

If you are working on IntelliJ IDE then you need to follow following given steps.

Step:1 Click on Project Structure contained in Left Pane of window and then right click on src, the menu list will be opened. Then select Open Module Settings option.
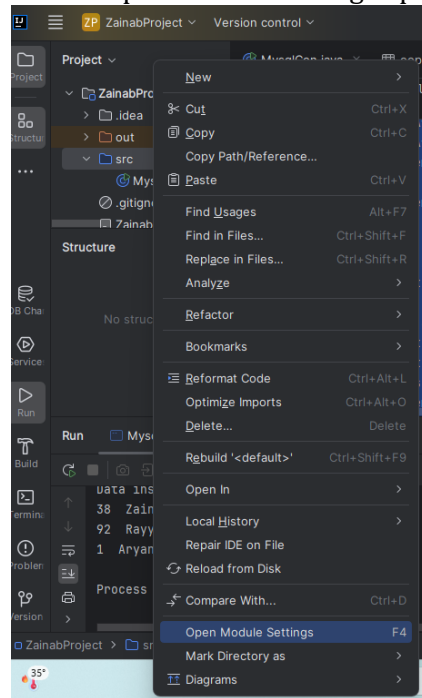

Step:2 Select Modules from Project Sttings and click on Dependencies tab and click on + and from which select JARs and Directories option.
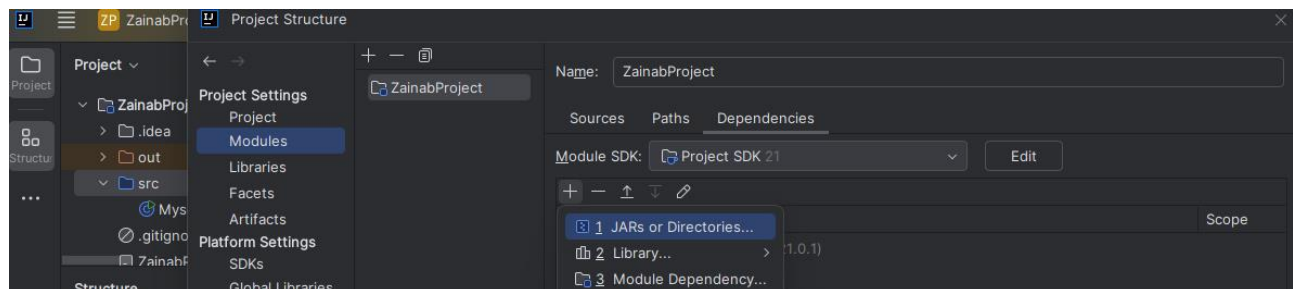
Step:3 Now another window will be opened where you need to browse the location where mysql connector jar file is located. Select that .jar file and click on OK button.
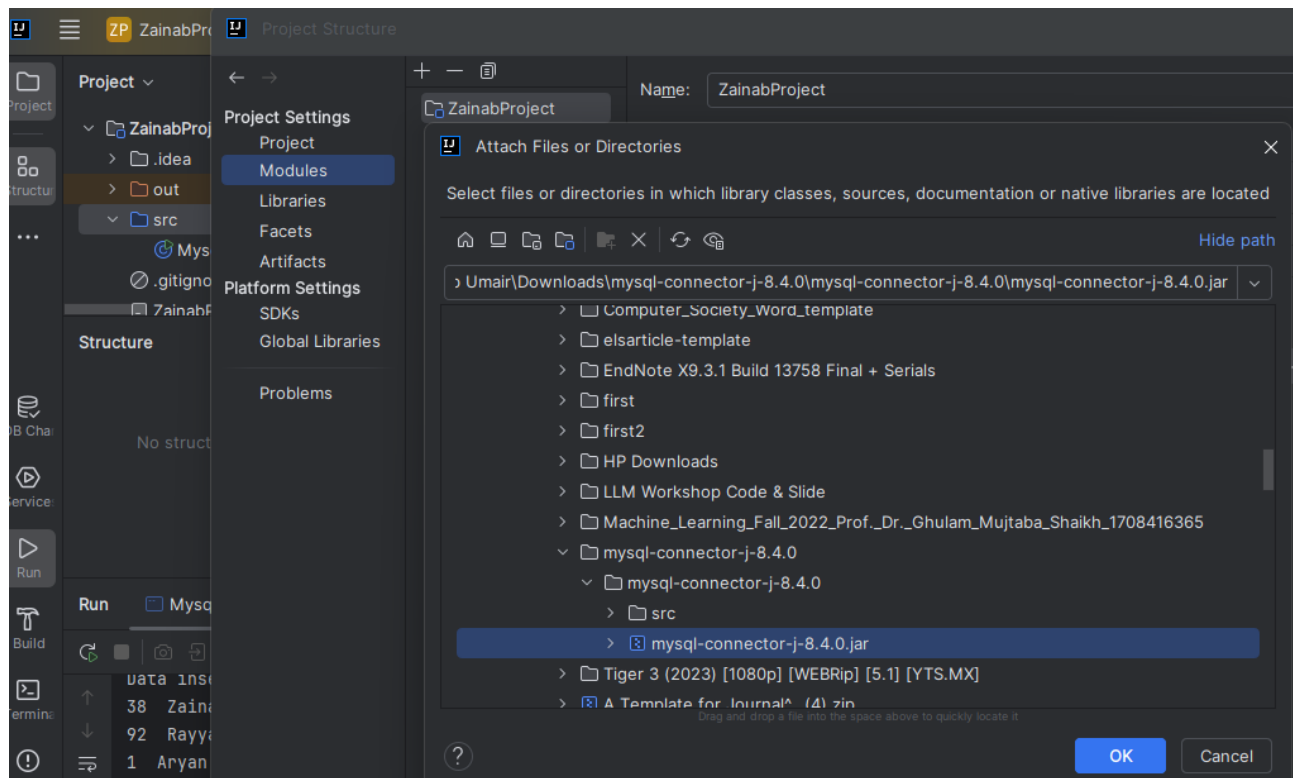
Step:4 Click on mysql-connector-j-8.4.0.jar checkbox and then click on Apply button and press OK button.
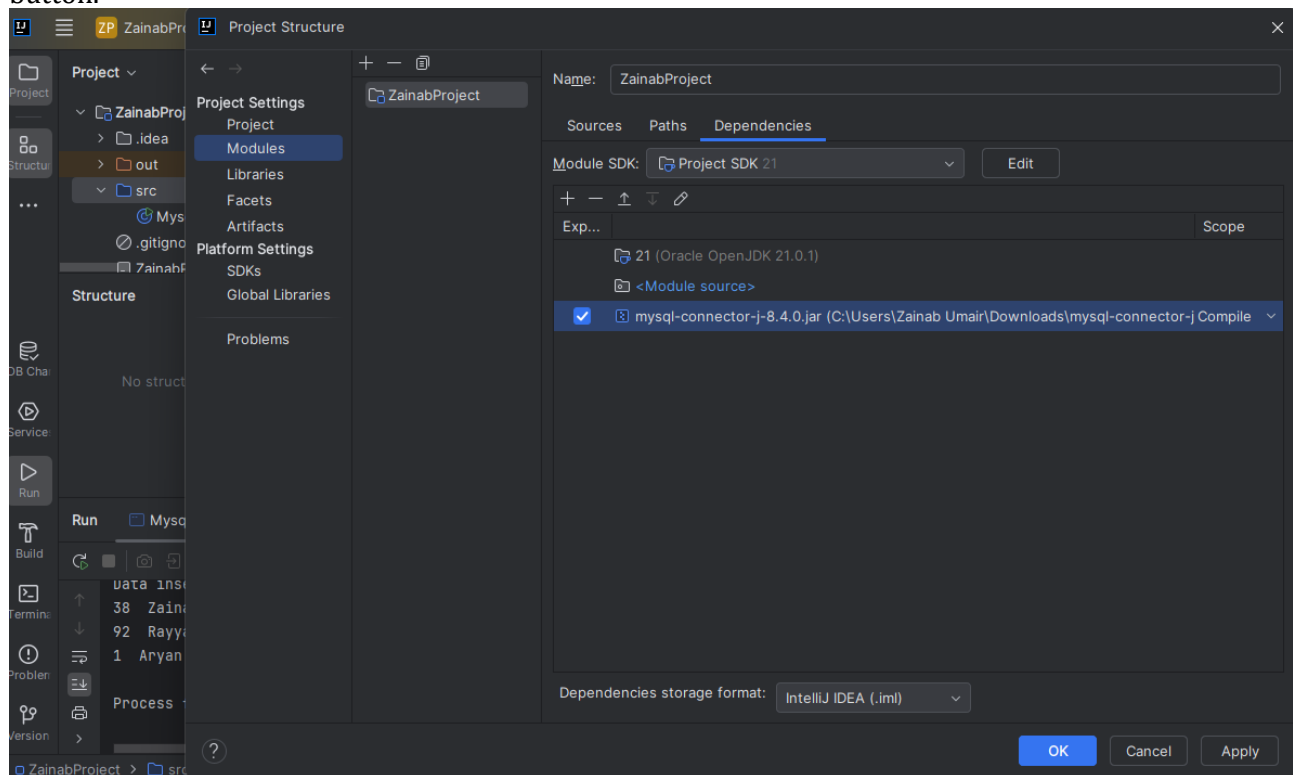
Inorder to know about how to connect and access databases in IntelliJ IDEA IDE follo the steps with the help of following instructions mentioned in link bellow:
https://www.youtube.com/watch?v=mpT95sa1zaE

Same example but now it is taking input from user:

```java
import java.sql.*;
import java.util.*;

public class MysqlCon {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/lab", "root", "12345");

            // Prepare SQL statement for insertion
            PreparedStatement statement = con.prepareStatement("INSERT INTO
emp VALUES (?, ?, ?)");

            // Input data from user
            System.out.print("Enter ID: ");
            int x = sc.nextInt();
            System.out.print("Enter Name: ");
            sc.nextLine();
            String y = sc.nextLine();
            System.out.print("Enter Age: ");
            int z = sc.nextInt();

            // Set parameters for prepared statement
            statement.setInt(1, x);
            statement.setString(2, y);
            statement.setInt(3, z);

            // Execute the statement to insert data
            int rowsAffected = statement.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println("Data inserted successfully.");
            } else {
                System.out.println("Failed to insert data.");
            }

            // Close resources
            statement.close();

            // Fetch and display all records from the database
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM emp");
            while (rs.next()) {
                System.out.println(rs.getInt(1) + "   " + rs.getString(2) + "
```
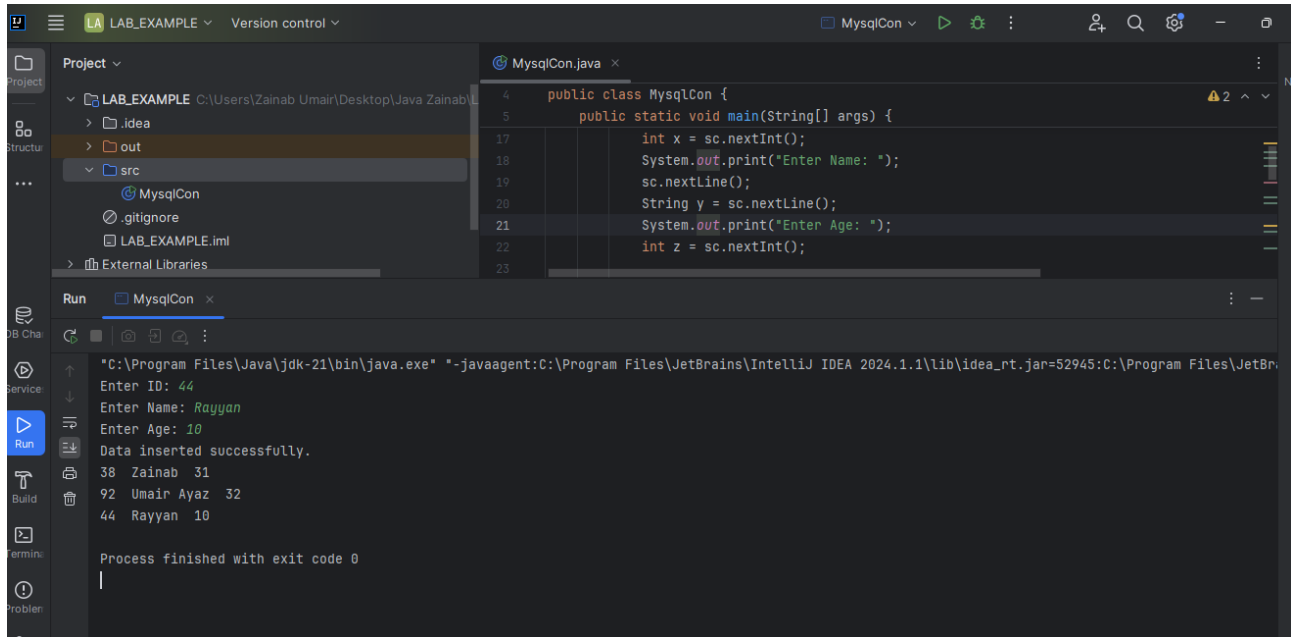
```
" + rs.getInt(3));
            }

            // Close connection
            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Database error: " + e.getMessage());
        }
    }
}
```


Exercise: Question:

Create a program to connect with MySQL database. You need to create a database and using that database, create a student table with (id, name, phone, semester, cgpa) attributes.

You need to create a GUI application where you need to add CRUD (Create, Read, Update & Delete) operations. All operations should be done with created student in database.