Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2024-25

Class:	**SE**	**Semester:**	**IV**
Course Code:	**CSL402**	**Course Name:**	**Database Management System Lab**

Name of Student:	**Shruti Gauchandra**
Roll No. :	**16**
Experiment No.:	**9**
Title of the Experiment:	**Demonstrate Database connectivity**
Date of Performance:	**06/03/25**
Date of Submission:	**13/03/25**

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Ms. Neha Raut

Signature :

Date:

Experiment No 9

Aim :- Write a java program to connect Java application with the MySQL database

Objective :- To learn database connectivity

Theory:

Database used : MySql

1. **Driver class:** The driver class for the mysql database is **com.mysql.jdbc.Driver**.
2. **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/loan management** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, can also use IP address, 3306 is the port number and loan management is the database name.
3. **Username:** The default username for the mysql database is Hiren.
4. **Password:** It is the password given by the user at the time of installing the mysql database. Password used is " ".

To connect a Java application with the MySQL database, follow the following steps.

- First create a database and then create a table in the mysql database.
- To connect java application with the mysql database, **mysqlconnector.jar** file is required to be loaded.
- download the jar file mysql-connector.jar
- add the jar file to the same folder as the java program.
- Compile and run the java program to retrieve data from the database.

Conclusion: Data has been retrieved successfully from a table by establishing database connectivity of java program with mysql database.

A)Explain steps to connect a java application with the MySQL database.

Step 1: Install MySQL and Configure Database

Ensure MySQL is installed and running. Create a database and table for your application.

Example SQL command to create a database and table:

```
CREATE DATABASE student_db;

USE student_db;

CREATE TABLE students (

id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(100),
```

```
age INT
);
```

Step 2: Add MySQL JDBC Driver to Java Project

Download the MySQL Connector/J (mysql-connector-java-x.x.x.jar) from the MySQL website.

Add it to your Java project:

- **If using Eclipse/IntelliJ, add it as an external library in the project settings.**
- **If using Maven, add this dependency in po.xml:**

```
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.33</version>
</dependency>
```

Step 3: Load the MySQL JDBC Driver

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
try {
Class.forName("com.mysql.cj.jdbc.Driver");
System.out.println("Driver Loaded Successfully!");
} catch (ClassNotFoundException e) {
e.printStackTrace();
}
```

Step 4: Establish a Connection to MySQL

```
String url = "jdbc:mysql://localhost:3306/student_db";
```

```
String user = "root";
String password = "yourpassword";
try {
Connection conn = DriverManager.getConnection(url, user, password);
System.out.println("Connected to MySQL Database!");
} catch (SQLException e) {
e.printStackTrace();
}
```

Step 5: Execute SQL Queries in Java

Once connected, use Statement or PreparedStatement to interact with the database.

Insert Data Example:

```
import java.sql.*;
public class InsertStudent {
public static void main(String[] args) {
String url = "jdbc:mysql://localhost:3306/student_db";
String user = "root";
String password = "yourpassword";
try {
Connection conn = DriverManager.getConnection(url, user, password);
String query = "INSERT INTO students (name, age) VALUES (?, ?)";
PreparedStatement pstmt = conn.prepareStatement(query);
pstmt.setString(1, "John Doe");
pstmt.setInt(2, 22);
pstmt.executeUpdate();
System.out.println("Data Inserted Successfully!");
conn.close();
} catch (SQLException e) {
e.printStackTrace();
```

```
  }
 }
}
```

Step 6: Retrieve Data from MySQL

```
String query = "SELECT * FROM students";
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
System.out.println("ID: " + rs.getInt("id") + ", Name: " + rs.getString("name") + ", Age: "+ rs.getInt("age"));
}
```

Step 7: Close the Connection

```
conn.close();
```