

Name	Chakkilala SuryaTeja Bharadwaj
Registration No.	2000030167
Course	Enterprise Programming
Section	1 – (Peer Mentor)
Title	JDBC CallableStatement in MySQL database with stored procedures
Date	8 <sup>th</sup> June 2022
GitHub Repository for Project	<a href="https://github.com/SuryaChakkilala/EP-Skill-1.git">https://github.com/SuryaChakkilala/EP-Skill-1.git</a>

## Question

### Skill 01: JDBC CallableStatement in MySQL database with stored procedures

- i. Create a table into the database by using the **CREATE TABLE** command for the below mentioned fields and create the stored procedure as mentioned below.
  - a. Create a table "**STUDENT**" in "**UNIVERSITY**" database that contains five columns and feed the 4 students records manually.

Fields	Datatype
regno	INT, PRIMARY KEY NOT NULL
firstname	VARCHAR(50)
lastname	VARCHAR(50)
CGPA	DOUBLE
BacklogsNo	INT
Contact_No	VARCHAR(50)

- b. Create Stored Procedure without Parameter for viewing the details of all students in the STUDENT Table named as "getDetails()"
  - c. Create Stored Procedure with an 'IN' Parameter of type INTEGER retrieve the student details of the particular students holding the specific regno named as "search(IN id INT)".
  - d. Create Stored Procedure with 6 'IN' parameters to insert the new record in to the STUDENT Table named as "insertData(IN id INT, IN fn VARCHAR(50), IN ln VARCHAR(50), IN cg DOUBLE, IN bl INT, IN cn VARCHAR(50))".
  - e. Create the Stored Procedure with a "OUT" parameter to display the number of students' data present in the STUDENT table named as "noofStud(OUT nos INT)".
  - f. Create the Stored Procedure with an "INOUT" parameter to retrieve the number of backlogs of a particular student holding the specific register number. NOTE: here need to use the single parameter named as "getBGL(INOUT info INT)" for sending the register number from JAVA Application to MySQL & retrieving the data from MySQL to JAVA application.
- ii. Create a Maven Project in Redhat Codeready Studio and implement the below mentioned operations in STUDENT table using switch case for selecting the respective operations:
  - a. Call all the created stored procedures through the options selected by the user using switch case.
  - b. Display the results & acknowledge the results of stored procedure.

Note: The students must implement all aforementioned operations and upload the PDF file with the MySQL Commands, Java Code, Screenshot of Output along with the Register Number, Name and Section Number of Practical/Skill. Plagiarism should be avoided to consent the grading.

#### Part(i): (MySQL Part)

- Database creation: **create database university;**
- Creating student table: **create table student(regno int primary key not null, firstname varchar(50), lastname varchar(50), cgpa double, backlogsno int, contact\_no varchar(50));**

```
mysql> create database university;
Query OK, 1 row affected (0.01 sec)

mysql> use university;
Database changed
mysql> create table student(regno int primary key not null, firstname varchar(50), lastname varchar(50), cgpa double, backlogsno int, contact_no varchar(50));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| student               |
+-----+
1 row in set (0.00 sec)
```

- Inserting records into student table: insert into student values(1, 'surya', 'chakkilala', 9.5, 0, '1111111111'), (2, 'prasanth', 'nalluri', 9.4, 0, '2222222222'), (3, 'sahithi', 'nethi', 9.4, 0, '3333333333'), (4, 'trishitha', 'ketineni', 9.5, 0, '4444444444'), (5, 'example', 'lastexample', 6.7, 2, '5555555555');

```
mysql> insert into student values(1, 'surya', 'chakkilala', 9.5, 0, '1111111111'), (2, 'prasanth', 'nalluri', 9.4, 0, '2222222222'), (3, 'sahithi', 'nethi', 9.4, 0, '3333333333'), (4, 'trishitha', 'ketineni', 9.5, 0, '4444444444'), (5, 'example', 'lastexample', 6.7, 2, '5555555555');
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from student;
+-----+-----+-----+-----+-----+-----+
| regno | firstname | lastname | cgpa | backlogsno | contact_no |
+-----+-----+-----+-----+-----+-----+
| 1 | surya | chakkilala | 9.5 | 0 | 1111111111 |
| 2 | prasanth | nalluri | 9.4 | 0 | 2222222222 |
| 3 | sahithi | nethi | 9.4 | 0 | 3333333333 |
| 4 | trishitha | ketineni | 9.5 | 0 | 4444444444 |
| 5 | example | lastexample | 6.7 | 2 | 5555555555 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- Creating getDetails() procedure:  
mysql> delimiter //  
mysql> create procedure getDetails()  
-> begin  
-> select \* from student;  
-> end; //

```
mysql> delimiter //
mysql> create procedure getDetails()
-> begin
-> select * from student;
-> end; //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> call getDetails();
-> //
```

```
+-----+-----+-----+-----+-----+-----+
| regno | firstname | lastname | cgpa | backlogsno | contact_no |
+-----+-----+-----+-----+-----+-----+
| 1 | surya | chakkilala | 9.5 | 0 | 1111111111 |
| 2 | prasanth | nalluri | 9.4 | 0 | 2222222222 |
| 3 | sahithi | nethi | 9.4 | 0 | 3333333333 |
| 4 | trishitha | ketineni | 9.5 | 0 | 4444444444 |
| 5 | example | lastexample | 6.7 | 2 | 5555555555 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> delimiter ;
```

- Creating search() procedure:  
mysql> delimiter //  
mysql> create procedure search(in id int)  
-> begin  
-> select \* from student where regno = id;  
-> end; //

```
mysql> delimiter //
mysql> create procedure search(in id int)
-> begin
-> select * from student where regno = id;
-> end; //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call search(4);
+-----+-----+-----+-----+-----+-----+
| regno | firstname | lastname | cgpa | backlogsno | contact_no |
+-----+-----+-----+-----+-----+-----+
| 4 | trishitha | ketineni | 9.5 | 0 | 4444444444 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

- Creating insertData() procedure:

```
mysql> delimiter //
mysql> create procedure insertData(in id int, in fn varchar(50), in ln varchar(50), in cg double, in bl int, in cn varchar(50))
-> begin
-> insert into student values(id, fn, ln, cg, bl, cn);
-> end; //
```

```
mysql> delimiter //
mysql> create procedure insertData(in id int, in fn varchar(50), in ln varchar(50), in cg double, in bl int, in cn varchar(50))
-> begin
-> insert into student values(id, fn, ln, cg, bl, cn);
-> end; //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call insertData(6, 'example2', 'lexample2', 7.77, 9, '6666666666');
Query OK, 1 row affected (0.00 sec)

mysql> call getDetails();
+-----+-----+-----+-----+-----+-----+
| regno | firstname | lastname | cgpa | backlogsno | contact_no |
+-----+-----+-----+-----+-----+-----+
| 1 | surya | chakkilala | 9.5 | 0 | 1111111111 |
| 2 | prasanth | nalluri | 9.4 | 0 | 2222222222 |
| 3 | sahithi | nethi | 9.4 | 0 | 3333333333 |
| 4 | trishitha | ketineni | 9.5 | 0 | 4444444444 |
| 5 | example | lastexample | 6.7 | 2 | 5555555555 |
| 6 | example2 | lexample2 | 7.77 | 9 | 6666666666 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

- Creating noofStud() procedure:

```
mysql> delimiter //
mysql> create procedure noofStud(out nos int)
-> begin
-> select count(*) from student into nos;
-> end; //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> delimiter //
mysql> create procedure noofStud(out nos int)
-> begin
-> select count(*) from student into nos;
-> end; //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call noofStud(@nos);
Query OK, 1 row affected (0.00 sec)

mysql> select @nos;
+-----+
| @nos |
+-----+
|    6 |
+-----+
1 row in set (0.00 sec)
```

- Creating getBGL() procedure:

```
mysql> delimiter //
mysql> create procedure getBGL(inout info int)
-> begin
-> select backlogsno into info from student where regno=info;
-> end; //
mysql> delimiter //
mysql> create procedure getBGL(inout info int)
-> begin
-> select backlogsno into info from student where regno=info;
-> end; //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> set @info = 6;
Query OK, 0 rows affected (0.00 sec)

mysql> call getBGL(@info);
Query OK, 1 row affected (0.00 sec)

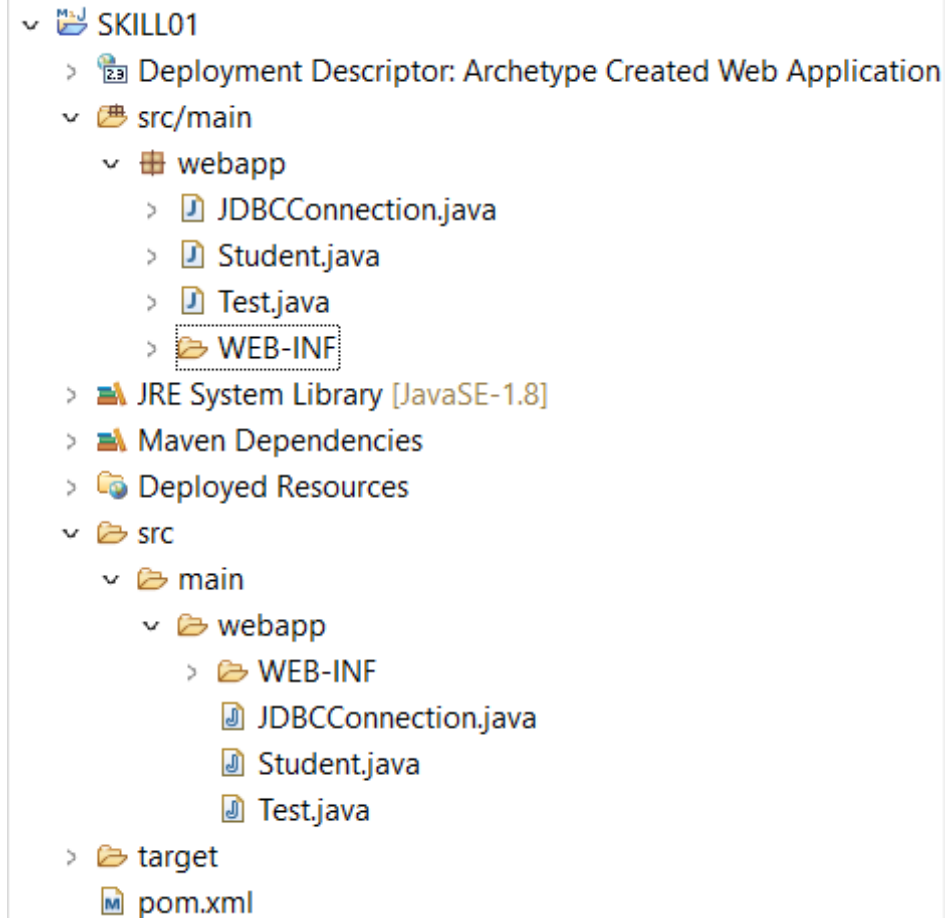
mysql> select @info;
+-----+
| @info |
+-----+
|    9 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SHOW PROCEDURE STATUS WHERE Db = 'university';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Db | Name | Type | Definer | Modified | Created | Security_type | Comment | character_set_client | collation_connection | Database Collation |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| university | getBGL | PROCEDURE | root@localhost | 2022-06-08 19:57:01 | 2022-06-08 19:57:01 | DEFINER | | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |
| university | getDetails | PROCEDURE | root@localhost | 2022-06-08 19:48:53 | 2022-06-08 19:48:53 | DEFINER | | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |
| university | insertData | PROCEDURE | root@localhost | 2022-06-08 19:56:14 | 2022-06-08 19:56:14 | DEFINER | | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |
| university | noofStud | PROCEDURE | root@localhost | 2022-06-08 19:56:42 | 2022-06-08 19:56:42 | DEFINER | | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |
| university | search | PROCEDURE | root@localhost | 2022-06-08 19:55:17 | 2022-06-08 19:55:17 | DEFINER | | cp850 | cp850_general_ci | utf8mb4_0900_ai_ci |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| student |
+-----+
1 row in set (0.00 sec)
```

## Part(ii): Java Program

## Folder Structure:



## Dependencies Tag in pom.xml:

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.29</version>
  </dependency>
</dependencies>
```

Program in next page →

**Student.java**

```
package webapp;
```

```
public class Student {
    private int regNo, backlogsNo;
    private String firstName, lastName, phoneNo;
    private double cgpa;

    public Student(int regNo, String firstName, String lastName, int backlogsNo, double cgpa,
String phoneNo) {
        this.regNo = regNo;
        this.firstName = firstName;
        this.lastName = lastName;
        this.backlogsNo = backlogsNo;
        this.cgpa = cgpa;
        this.phoneNo = phoneNo;
    }

    public void setRegNo(int regNo) {
        this.regNo = regNo;
    }

    public int getRegNo() {
        return this.regNo;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getFirstName() {
        return this.firstName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getLastName() {
        return this.lastName;
    }

    public void setBacklagsNo(int backlogsNo) {
        this.backlogsNo = backlogsNo;
    }

    public int getBacklogsNo() {
        return this.backlogsNo;
    }

    public void setCgpa(double cgpa) {
        this.cgpa = cgpa;
    }

    public double getCgpa() {
        return this.cgpa;
    }

    public void setPhoneNo(String phoneNo) {
        this.phoneNo = phoneNo;
    }
}
```

```

    public String getPhoneNo() {
        return this.phoneNo;
    }

    public String toString() {
        return "RegNo: " + this.regNo + "\tFirst Name: " + this.firstName + "\tLast Name: " +
this.lastName + "\tBacklogs No: " + this.backlogsNo + "\tCGPA: " + this.cgpa + "\tPhone No: " +
this.phoneNo;
    }
}

```

### JDBCConnection.java

```
package webapp;
```

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

```

```
public class JDBCConnection {
```

```

    public static List<Student> getStudents() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/university?characterEncoding=latin1&useConfigs=maxPerformance",
            "root", "SuryaTeja@9192");
            CallableStatement stmt = con.prepareCall("{call getDetails()}");
            ResultSet rs = stmt.executeQuery();
            List<Student> studentsList = new ArrayList<Student>();
            while(rs.next()) {
                studentsList.add(new Student(rs.getInt(1), rs.getString(2), rs.getString(3),
                    rs.getInt(5), rs.getDouble(4), rs.getString(6)));
            }
            return studentsList;
        } catch(Exception e) {
            System.out.println(e);
        }
        return new ArrayList<Student>();
    }

```

```

    public static int insertStudent(Student student) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/university?characterEncoding=latin1&useConfigs=maxPerformance",
            "root", "SuryaTeja@9192");
            CallableStatement ps = con.prepareCall("{call insertData(?, ?, ?, ?, ?, ?)}");
            ps.setInt(1, student.getRegNo());
            ps.setString(2, student.getFirstName());
            ps.setString(3, student.getLastName());
            ps.setDouble(4, student.getCgpa());
            ps.setInt(5, student.getBacklogsNo());

```

```

        ps.setString(6, student.getPhoneNo());
        return ps.executeUpdate();
    } catch (Exception e) {
        System.out.println(e);
    }
    return 0;
}

public static int getBGL(int id) {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/university?characterEncoding=latin1&useConfigs=maxPerformance",
            "root", "SuryaTeja@9192");
        CallableStatement ps = con.prepareCall("{call getBGL(?)}");
        ps.registerOutParameter(1, java.sql.Types.INTEGER);
        ps.setInt(1, id);
        ps.executeUpdate();
        return ps.getInt(1);
    } catch (Exception e) {
        System.out.println(e);
    }
    return -1;
}

public static int noOfStudents() {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/university?characterEncoding=latin1&useConfigs=maxPerformance",
            "root", "SuryaTeja@9192");
        CallableStatement ps = con.prepareCall("{call noofStud(?)}");
        ps.registerOutParameter(1, java.sql.Types.INTEGER);
        ps.executeUpdate();
        return ps.getInt(1);
    } catch (Exception e) {
        System.out.println(e);
    }
    return -1;
}

public static Student search(int id) {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/university?characterEncoding=latin1&useConfigs=maxPerformance",
            "root", "SuryaTeja@9192");
        CallableStatement ps = con.prepareCall("{call search(?)}");
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        if(rs.next()) {
            return new Student(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(5),
rs.getDouble(4), rs.getString(6));
        }
        return null;
    } catch (Exception e) {
        System.out.println(e);
    }
}

```



```
        return null;
    }
}
```

### Test.java

```
package webapp;

import java.util.Iterator;
import java.util.Scanner;

public class Test {
    private static final Scanner s = new Scanner(System.in);

    private static void displayMenu() {
        System.out.println("MENU");
        System.out.println("1. Get All Students Details");
        System.out.println("2. Search for Student by ID");
        System.out.println("3. Insert a Student record");
        System.out.println("4. Get the Number of records");
        System.out.println("5. Get backlogs of a student with ID");
    }

    private static int getChoice() {
        displayMenu();
        System.out.println("Enter your choice: ");
        int choice = s.nextInt();
        return choice;
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        boolean isRunning = true;
        while(isRunning) {
            int id = 0;
            switch(getChoice()) {
                case 1:
                    Iterator<Student> it = JDBCConnection.getStudents().iterator();
                    System.out.println("Student Details: ");
                    while(it.hasNext()) {
                        System.out.println(it.next());
                    }
                    System.out.println();
                    break;
                case 2:
                    System.out.print("Enter the ID: ");
                    id = s.nextInt();
                    System.out.println(JDBCConnection.search(id));
                    break;
                case 3:
                    System.out.print("Enter ID: ");
                    id = s.nextInt();
                    System.out.print("Enter First Name: ");
                    String firstName = s.next();
                    System.out.print("Enter Last Name: ");
                    String lastName = s.next();
                    System.out.print("Enter CGPA: ");
                    double cgpa = s.nextDouble();
                    System.out.print("Enter Backlog Count: ");
            }
        }
    }
}
```

```
        int backlogs = s.nextInt();
        System.out.print("Enter phone number: ");
        String phoneNo = s.next();
        int count = JDBCConnection.insertStudent(new Student(id, firstName,
lastName, backlogs, cgpa, phoneNo));
        if(count == 1) {
            System.out.println("Insertion Successful");
        } else {
            System.out.println("Insertion failed");
        }
        break;
    case 4:
        System.out.println("No. of records: " + JDBCConnection.noOfStudents());
        break;
    case 5:
        System.out.println("Enter the ID to find the backlogs: ");
        id = s.nextInt();
        System.out.println("Backlog Count for " + id + ": " +
JDBCConnection.getBGL(id));
        break;
    default:
        System.out.println("Stopped.");
        isRunning = true;
    }
}
s.close();
}
```

**-----THE END-----**