Laboratory Work

Subject: Java Technologies
Branch: B.Tech. (CE)
Semester: IV
Batch: A1

Student Roll No: CE001
Student Name: NISARG KALPESHBHAI AMLANI

Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad – 387001.
Gujarat, INDIA.

Lab 6

Question 1)

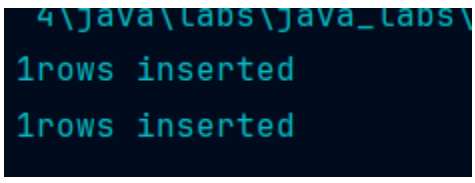
1. Write a Java application to perform operations for student information like (id[Primary key, Auto increment],firstName, lastName, branch, username and password) from a database using JDBC.
    - Insert two records for student
    - Practice the use of the following methods of the ResultSet interface: absolute(), afterLast(), beforeFirst(), first(), isFirst(), isLast(), last(), previous(), next(), relative().

Solution)

```
import java.sql.*;
public class StatementCRUD {
   public static void main(String[] args) {
      // TODO Auto-generated method stub
      try (Connection con =

DriverManager.getConnection("jdbc:mysql://localhost:3306/java","root"
, "")) {
         Statement s ;
         s = con.createStatement();
         String query =
```

```
"INSERT INTO `student` (`firstName`,`lastName`,`branch`,`username`,`password`) VALUES ('Nisarg','Amlani','ce','djdjsfdjhfj','jdshfjdh')";

int i = s.executeUpdate(query);

        System.out.println(i + "rows inserted");
                String query2 = "INSERT INTO `student` (`firstName`,`lastName`,`branch`,`username`,`password`) VALUES ('Nisarg1','Amlani','ce','djdjsfdjhfj','jdshfjdh')";

        i = s.executeUpdate(query2);
        System.out.println(i + "rows inserted");
      }
      catch (SQLException e) {
        System.out.println(e.getMessage());
      }
   }
}
```

Screenshot)



Question 2)

Using JDBC API and MySql database perform the following operations.

I. create a table MOVIES with following columns in the database:
Id of type **INTEGER AUTO INCREMENT**,
Title of type **VARCHAR (50)**,
Genre of type **VARCHAR (50)**,
YearOfRelease of type **INTEGER**.

II. Define **Movie** class with following data members
private Integer id;
private String title;
private String genre;
private Integer yearOfRelease;
Create getters and setters for the mentioned data members.

III. Define following methods in a class, test the methods according to user input

A. **createMovie(Movie m)**- it will insert a new record for a movie.
B. **deleteMovie(int MovieID)**- it will delete a movie with given MovieID
C. **updateMovieTitle(String title, int id)**- it will update the title of a movie with given id.
D. **findMovieById(int MovieId)**- it will display all details of a movie with a given MovieId
E. **findAllMovie()**- it will display all details of all movies

Solution)

Movie Class : -

```
class Movie{
    private Integer id;
    private String title;
    private String gener;
    private String yearOfRelease;

    public String getYearOfRelease() {
        return yearOfRelease;
    }
    public void setYearOfRelease(String yearOfRelease) {
        this.yearOfRelease = yearOfRelease;
    }
    public String getTitle() {
        return title;
    }
    public String getGener() {
        return gener;
    }
    @Override
    public String toString() {
        return "Movie{" +
                "id=" + id +
                ", title='" + title + '\'' +
```

```
            ", gener='" + gener + '\'' +
            ", yearOfRelease=" + yearOfRelease +
            '}';
}

public void setGener(String gener) {
    this.gener = gener;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public void setTitle(String title) {
    this.title = title;
}

void createMovie(Movie m)
{
```

```
        try(Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java","
root",""))
        {
            Statement s = con.createStatement();
            String query;
            query = "INSERT INTO movies (title,genre,yearofrelease)
VALUES
('"+m.getTitle()+"','"+m.getGener()+"','"+m.getYearOfRelease()+"')";
            int i = s.executeUpdate(query);
            System.out.println(i + " rows inserted");
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }

    void deleteMovie(Integer id)
    {
        try(Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java","root"
,"")){
```

```
            Statement s = con.createStatement();
            String query;
            query = "DELETE FROM movies WHERE id =  '"+id+"'";
            int i = s.executeUpdate(query);
            System.out.println(i + " rows deleted");
        }
        catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    void updateMovieTitle(Integer id,String title)
    {
        try(Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java","root"
,""))
        {
            Statement s = con.createStatement();
            String query ;
            query = "UPDATE movies   SET title ='"+title+"' WHERE id    =
"+id+"' '";
            int i = s.executeUpdate(query);
            System.out.println(i + " rows updated");
```

```
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
    void findMovieByid(Integer id)
    {
        try(Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java","root"
,""))
        {
            Statement s = con.createStatement();
            String query ;
            query = "SELECT * FROM movies WHERE id ='"+id+"' ";
            ResultSet rs = s.executeQuery(query);
            while(rs.next())
            {
                System.out.println("id :- " +  rs.getInt("id")+ "\ntitle :-
"+rs.getString("title")+"\ngenre :- "+rs.getString("genre")+"\nYear :-
"+rs.getString("yearofrelease"));
            }
```

```
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }

    void findallMovie()
    {
        try(Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java","root"
,""))
        {
            Statement s = con.createStatement();
            String query ;
            query = "SELECT * FROM movies";
            ResultSet rs = s.executeQuery(query);
            while(rs.next())
            {
                System.out.println("id :- " +  rs.getInt("id")+ "\ntitle :-
"+rs.getString("title")+"\ngenre :- "+rs.getString("genre")+"\nYear :-
"+rs.getString("yearofrelease"));
```

```
System.out.println("-------------------------------------------------");

            }


        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

Main Class :-

```
class Main{
    public static void main(String[] args) {
        Movie m = new Movie();
        m.setTitle("The Dark Knight");
        m.setGener("Action");
        m.setYearOfRelease("2008");
        m.createMovie(m);
        m.deleteMovie(1);
```

```
        m.updateMovieTitle(2,"The Dark Knight Rises");
        m.findMovieByid(2);
        m.findallMovie();

    }
}
```

ScreenShots)

Create Movie :-

Console

```
1 rows inserted
```

Database

WHERE ORDER BY

	id	Title	Genre	YearOfRelease
1	3	JOKER	Action	2008
2	4	The Martian	Sci-Fi	2020
3	5	The Dark Knight	Action	2008

Delete Movie :-

Console

```
1 rows deleted
```

Database

	id	Title	Genre	YearOfRelease
1	3	JOKER	Action	2008
2	4	The Martian	Sci-Fi	2020

Update Movie : -

Console

```
1 rows updated
```

DataBase

	id	Title	Genre	YearOfRelease
1	3	The Dark Knight Ri…	Action	2008
2	4	The Martian	Sci-Fi	2020

Find Movie by id :-

```
id :- 3
title :-  The Dark Knight Rises
genre :- Action
Year :- 2008
```

Find All Movie :-

```
id :- 3
title :-  The Dark Knight Rises
genre :- Action
Year :- 2008
-----------------------------------------------------
id :- 4
title :-  The Martian
genre :- Sci-Fi
Year :- 2020
-----------------------------------------------------
```

Question 3)

Create a Generic class Calculator which can perform addition, subtraction, multiplication and division. Make sure that Calculator class works for Numeric values only. Write an appropriate main method in TestCalculator class.

Solution)

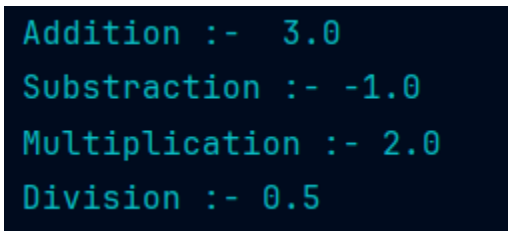
Calculator Class :-

```
public class Calculator<T extends Number> {
    public double add(T a, T b) {
        return a.doubleValue() + b.doubleValue();
    }

    public double subtract(T a, T b) {
        return a.doubleValue() - b.doubleValue();
    }

    public double multiply(T a, T b) {
        return a.doubleValue() * b.doubleValue();
    }

    public double divide(T a, T b) {
        if (b.doubleValue() == 0) {
            throw new IllegalArgumentException("Cannot divide by zero");
        }
        return a.doubleValue() / b.doubleValue();
    }
}
```

Main Class :-

```
public class calc_main {
    public static void main(String[] args) {
        Calculator<Integer> calc = new Calculator<>();
        System.out.println(calc.add(1, 2));
        System.out.println(calc.subtract(1, 2));
        System.out.println(calc.multiply(1, 2));
        System.out.println(calc.divide(1, 2));
    }
}
```

Screen Shots)



Question 4)

Write a Java program to create a generic method that takes two arrays of T type and checks if they have the same elements in the same order.

```
public class CheckArr {

    static <T>  boolean check(T[] arr1, T[] arr2) {
        if (arr1.length != arr2.length) {
            return false;
        }
        for (int i = 0; i < arr1.length; i++) {
            if (!arr1[i].equals(arr2[i])) {
                return false;
            }
        }
        return true;
    }
    public static void main(String[] args) {
        Integer[] array1 = {1, 2, 3, 4, 5};
        Integer[] array2 = {1, 2, 3, 4, 5};
        Integer[] array3 = {1, 2, 3, 4, 6};
```

```
        System.out.println(check(array1, array2));  // Output: true
        System.out.println(check(array1, array3));  // Output: false
    }
}
```

Screen Shot)