Experiment 4

Java Database connection testing using Result Set

Name: Sai Harsha Vardhan AVN

Roll-no: BCSE1823

Batch: 'A'

Aim: Create &connect to database, use SQL for ResultSet

Theory:

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database.

Steps for JDBC connection are as follows:

- Import JDBC packages.
- Load and register the JDBC driver.
- Open a connection to the database.
- Create a statement object to perform a query.
- Execute the statement object and return a query resultset.
- Process the resultset.
- Close the resultset and statement objects.
- Close the connection.

JDBC - Java Database Connectivity
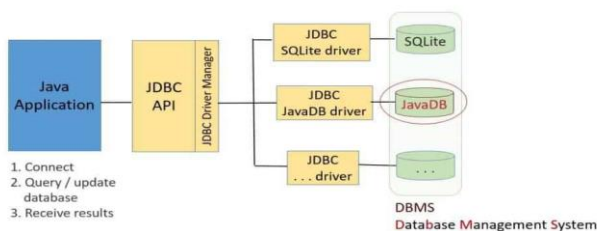
Code:

```java
import java.io.*;

import java.sql.*;

public class ConnectionofJDBC {

    public static void main(String args[]){

    InputStreamReader isr = new InputStreamReader(System.in);

    BufferedReader br = new BufferedReader(isr);

    String firstname,lastname,email;

    int id,cnt=0;

    Connection con;

    PreparedStatement pst;

    ResultSet rs;

try{

    Class.forName("com.mysql.jdbc.Driver");

    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?useSSL=false","root","mysql");

    pst = con.prepareStatement("select * from leads1 where id=?")

    do{

        System.out.println("Enter id");

        id = Integer.parseInt(br.readLine());

        pst.setInt(1,id);

        rs = pst.executeQuery();
```

```java
    cnt = 0;

        while(rs.next())

    {

        cnt++;

        System.out.println(rs.getString(2)+"\t"+rs.getString(4));

    }

        if(cnt ==0 && id>0)

        System.out.println("Record dont existes");

    }while(id>0);

System.out.println(" reords searched Successfully");

pst.close();

con.close();

}

catch(Exception ex){

 System.out.println(ex);

 }

}

}
```
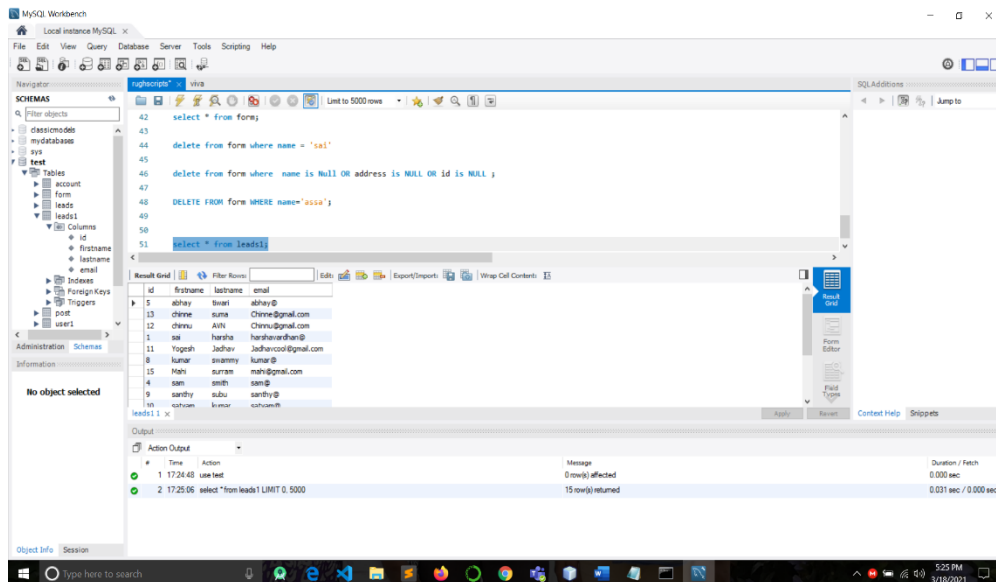
Output:

1) List of record's already which are available in a database:

2) Retrieving the entries using with the help of ResultSet:

Conclusion: Connection creation for SQL is successfully established.