Experiment No. 4

Aim of the Experiment: Write a program to insert and retrieve the data from the database using JDBC.

Objective:

To create database using MySQL to insert and retrieve the data from the database using JDBC.

Resources: MySQL, MySQL connector, Eclipse IDE 2018, JDK 1.8.0 is required

Course Outcome Addressed: CO4

Theory:

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as −

- Java Applications
- Java Applets
- Java Servlets
- Java ServerPages (JSPs)
- Enterprise JavaBeans (EJBs).

All of these different executables are able to use a JDBC driver to access a database, and take advantage of the stored data.

JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

Structured Query Language (SQL) is a standardized language that allows you to perform operations on a database, such as creating entries, reading content, updating content, and deleting entries.

SQL is supported by almost any database you will likely use, and it allows you to write database code independently of the underlying database

MySQL is the most popular Open Source Relational SQL database management system.

To establish a connection with MySQL in JDBC, first we have to install MySQL on our system and we have to add the **MySQL connector** (it is a .jar file containing the classes that are the implementation of interfaces provided by Sun Microsystems) to our class path variable.

Download mysql-connector.jar

https://static.javatpoint.com/src/jdbc/mysql-connector.jar

- To add this to eclipse java project,
- Right click on Project and go to properties
- Click on Java Build Path -> Libraries -> Classpath -> Add External JARs
- Provide the path of mysql-connector.jar file and click on Open
- Click Apply and Close

To connect Java application with the MySQL database, we need to follow 5 following steps. We are using MySql as the database; so we need to know following information's for the Mysql database:

- Driver class: The driver class for the mysql database is com.mysql.jdbc.Driver
- Connection URL: The connection URL for the mysql database is jdbc:mysql://localhost:3306/sppu 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 sppu is the database name. We may use any database, in such case, we need to replace the sppu with our database name.
- Username: The default username for the mysql database is root.
- 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.

VIMP Classes Use in JDBC Practical:

- DriverManager Class: DriverManager class acts as an interface between users and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver.
- Connection Interface: A Connection is a session between a Java application and a database. It helps to establish a connection with the database. The Connection interface is a factory of Statement, PreparedStatement.
- Statement Interface: The Statement interface provides methods to execute queries with the database. The statement interface is a factory of ResultSet i.e. it provides factory method to get the object of ResultSet.
- ResultSet Interface: The object of ResultSet maintains a cursor pointing to a row of a table. Initially, cursor points to before the first row.
- PreparedStatement Interface: The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query.
-

SOURCE CODE:

4a. Write a program to insert the data from the database using JDBC

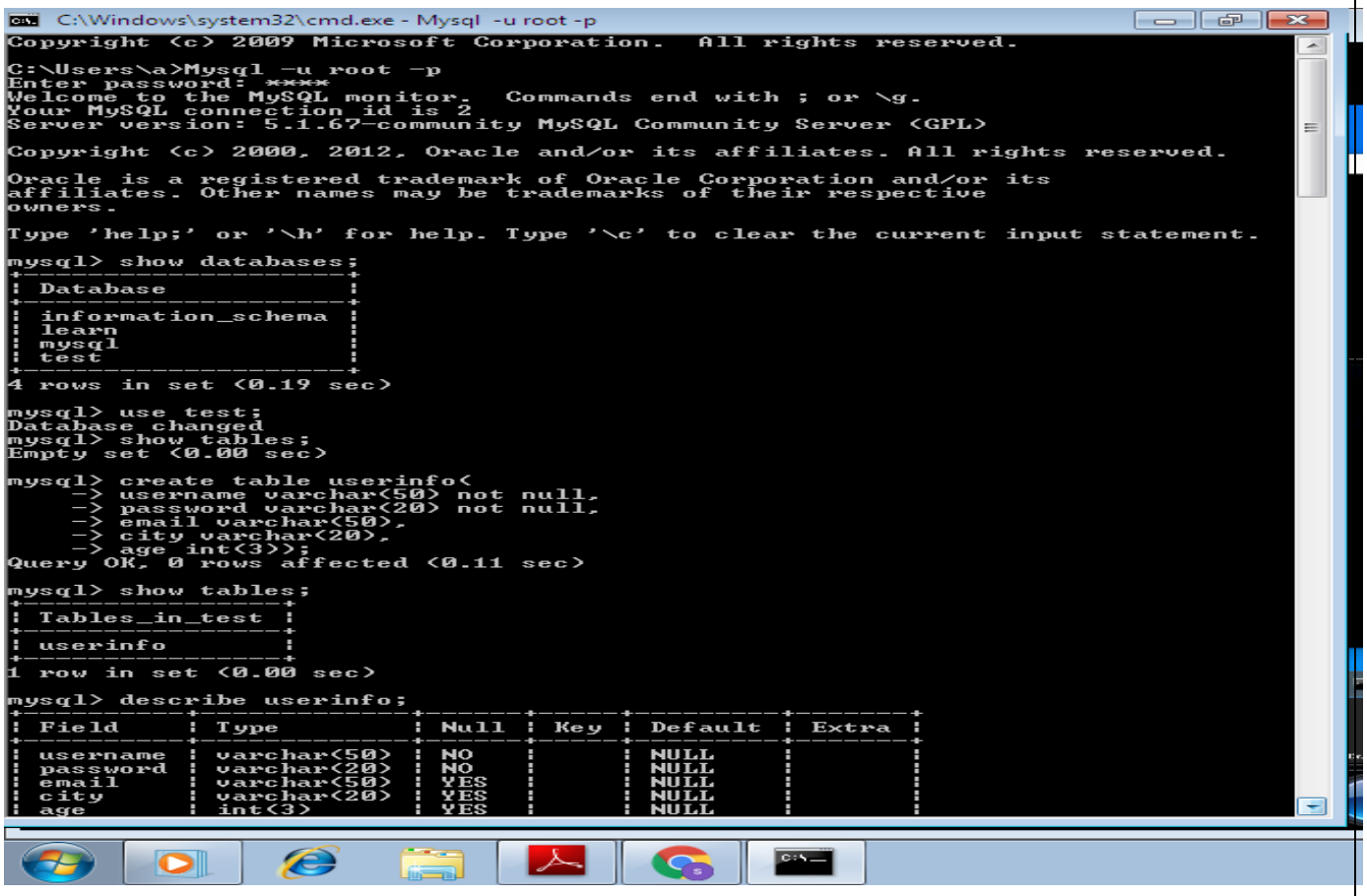
Program:

```
package Mysql;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;


public class SQLPreparedStatementInsert {
public static void main(String[] args) {
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/test?characterEncoding=latin1","root","ro
ot");
```

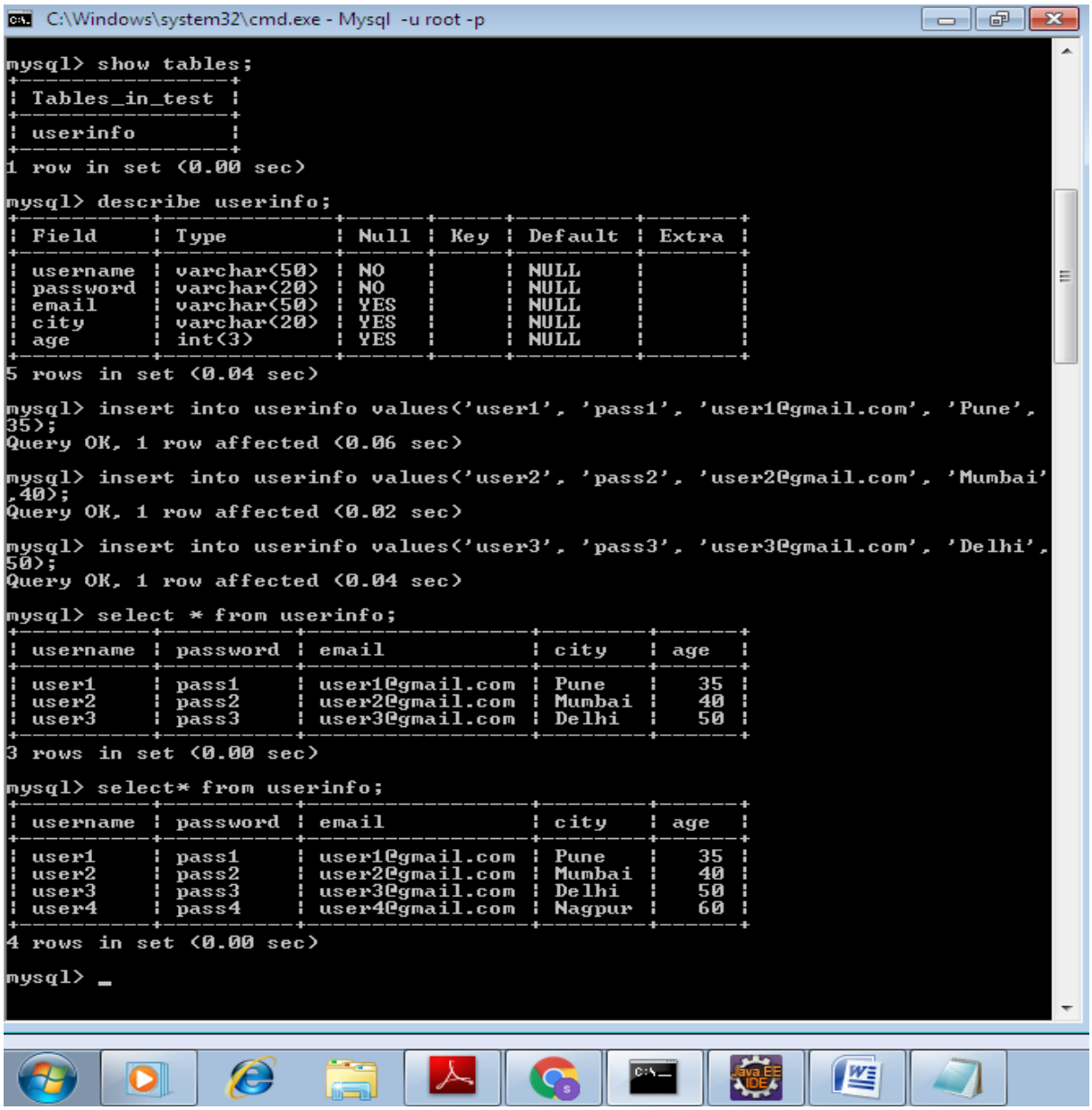
```
PreparedStatement stmt = con.prepareStatement("insert into userinfo
values (?,?,?,?,?)");
stmt.setString(1,"user4");
stmt.setString(2,"pass4");
stmt.setString(3,"user4@gmail.com");
stmt.setString(4, "Nagpur");
stmt.setString(5, "60");

int i = stmt.executeUpdate();
System.out.println(i + "Records inserted..");
con.close();
}
catch(Exception e){
System.out.println(e);
}
}
}
```

Output:

Before Insert program-

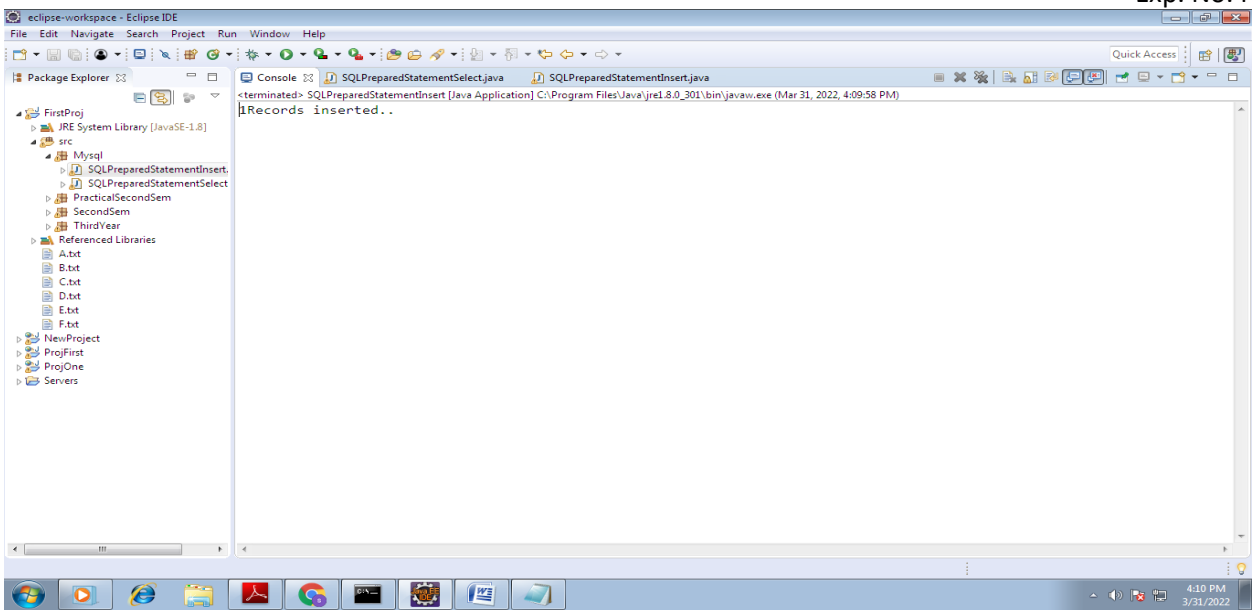


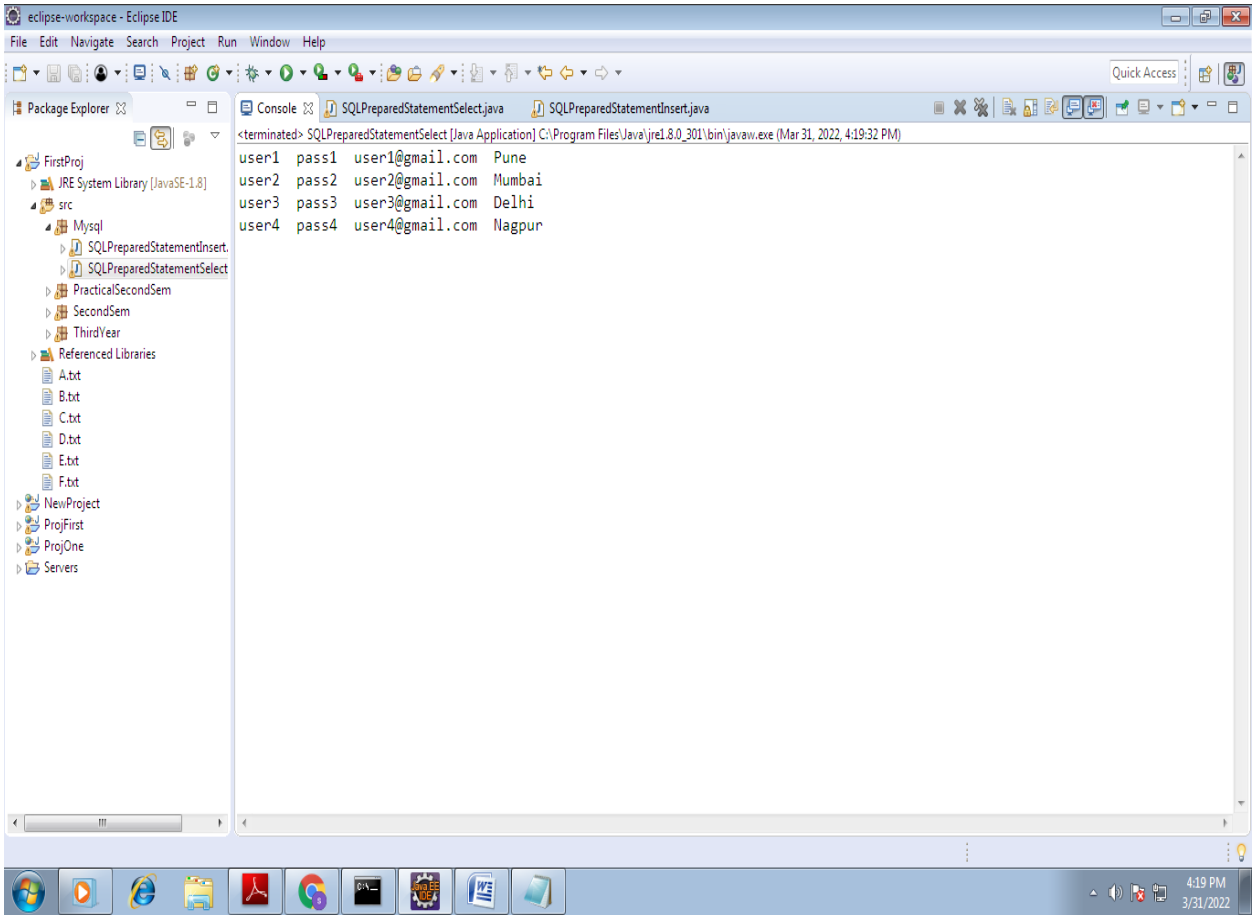
After Insert program-



Subject: Advanced Java Programming Lab (Elective - II) (304198)(C)

Exp. No.4



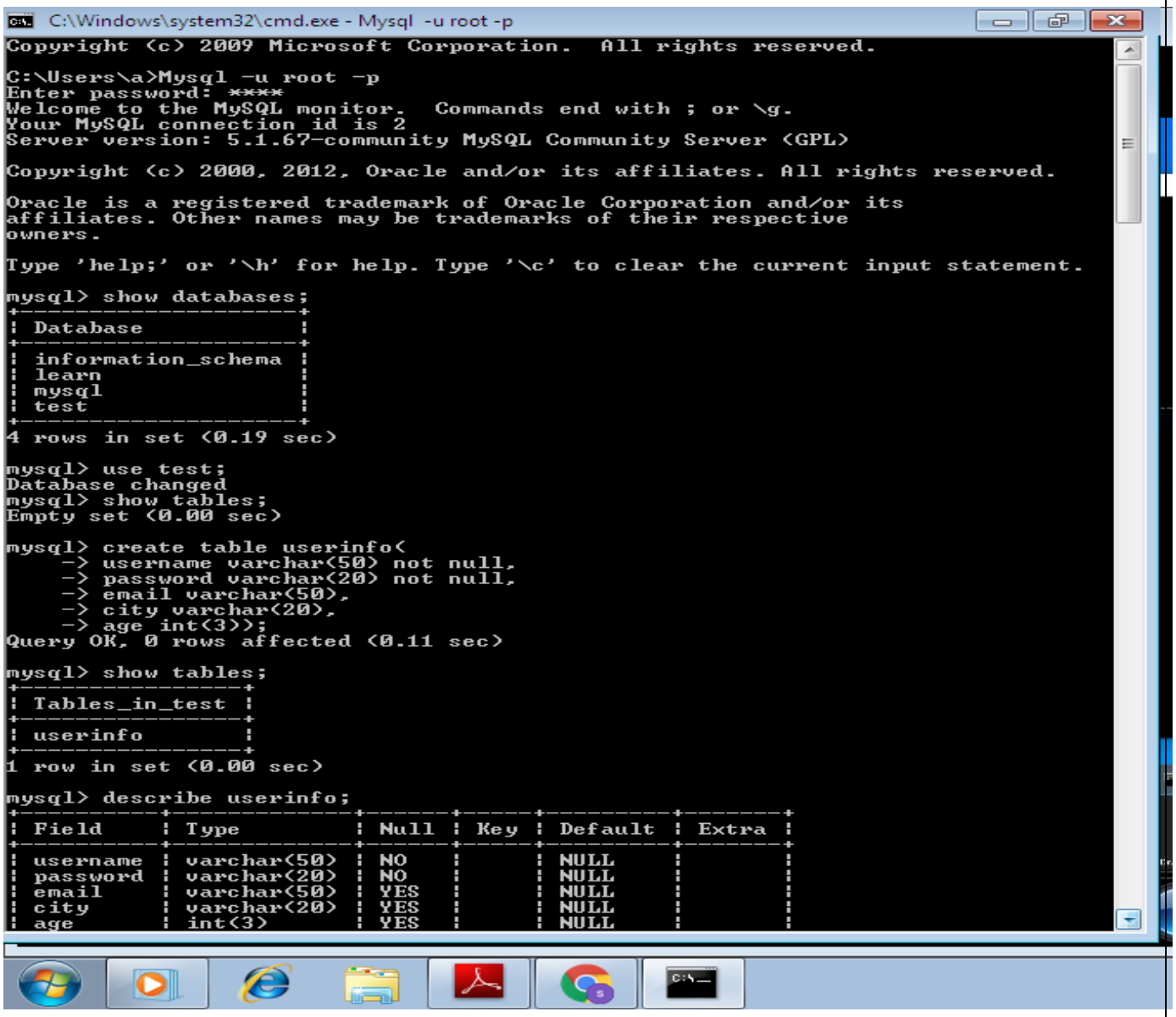


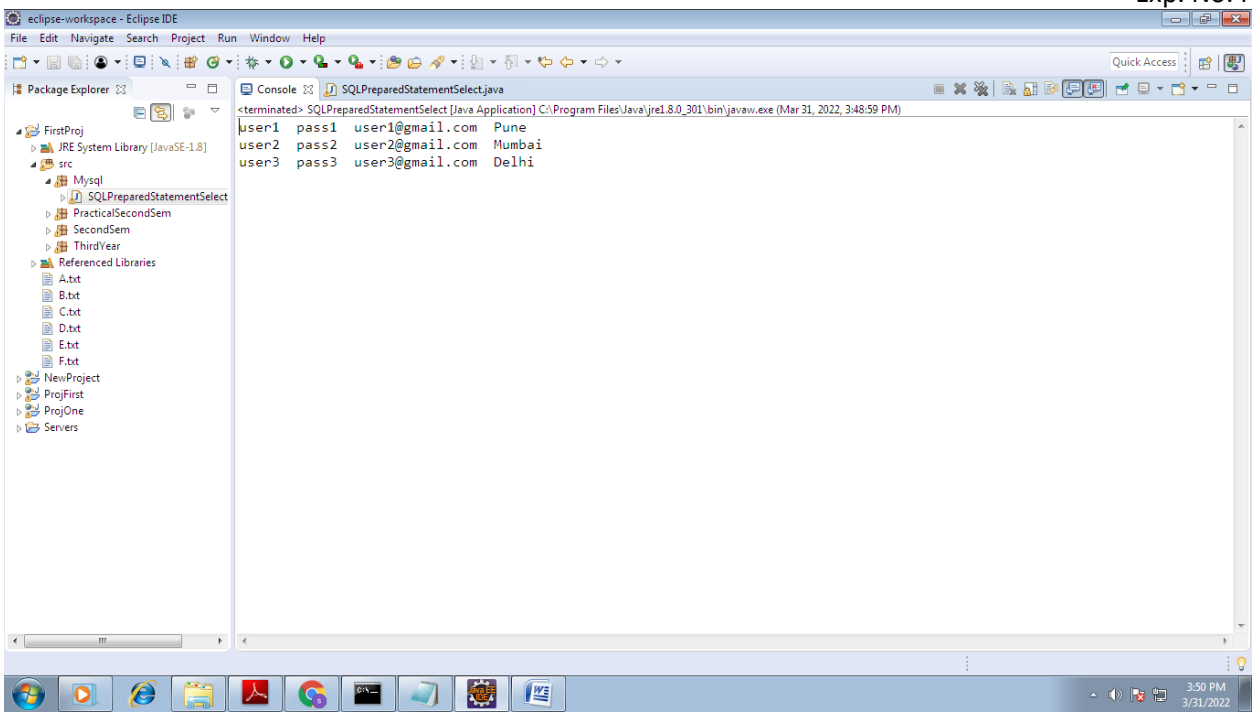
AJP Lab 4b Write a program to retrieve the data from the database using JDBC.

Program:

```
package Mysql;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
public class SQLPreparedStatementSelect {

public static void main(String[] args) {
try{

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

Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/test?characterEncoding=latin1","root","ro
ot");


PreparedStatement stmt = con.prepareStatement("select * from userinfo");
ResultSet rs = stmt.executeQuery();
while(rs.next())
{

System.out.println(rs.getString(1) + "   " +
rs.getString(2) + "   " + rs.getString(3) + "   " +
rs.getString(4) + "   ");
}

con.close();
}

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

Output:


C:\Windows\system32\cmd.exe - Mysql -u root -p
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\a>Mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.67-community MySQL Community Server (GPL)
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.
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> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| learn |
| mysql |
| test |
+--------------------+
4 rows in set (0.19 sec)
mysql> use test;
Database changed
mysql> show tables;
Empty set (0.00 sec)
mysql> create table userinfo(
-> username varchar(50) not null,
-> password varchar(20) not null,
-> email varchar(50),
-> city varchar(20),
-> age int(3));
Query OK, 0 rows affected (0.11 sec)
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| userinfo |
+----------------+
1 row in set (0.00 sec)
mysql> describe userinfo;
+----------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
username	varchar(50)	NO		NULL	
password	varchar(20)	NO		NULL	
email	varchar(50)	YES		NULL	
city	varchar(20)	YES		NULL	
age	int(3)	YES		NULL	



Conclusion:

References:

Herbert Schildt , "Java : The Complete Reference" Tata McGraw-Hill (7th Edition).

Questions:

1. What is JDBC?
2. What is JDBC Driver?
3. What are the steps to connect to the database in java?
4. What are the JDBC API components?
5. What are the differences between execute, executeQuery, and executeUpdate?