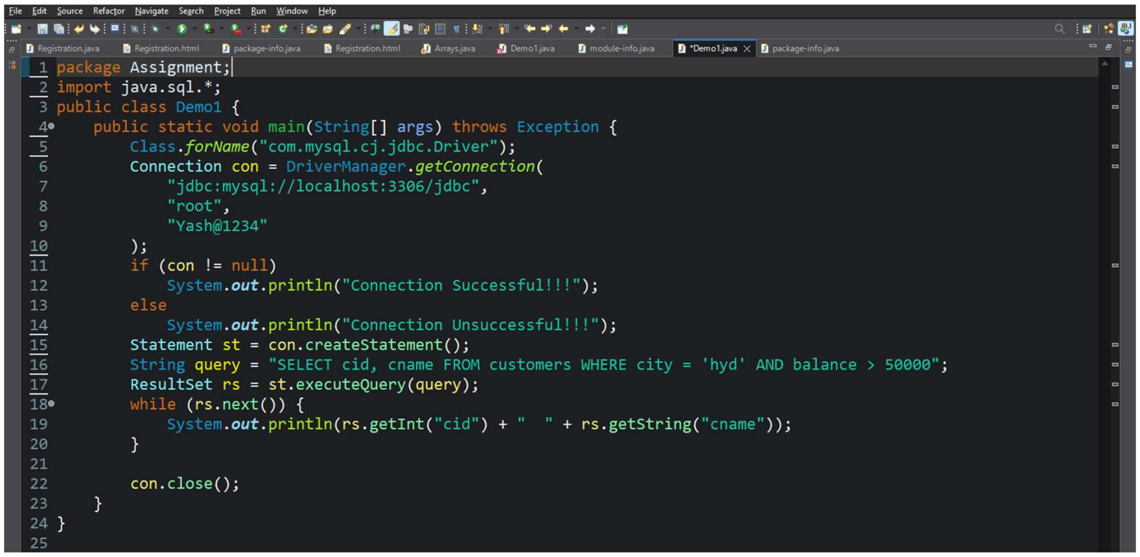
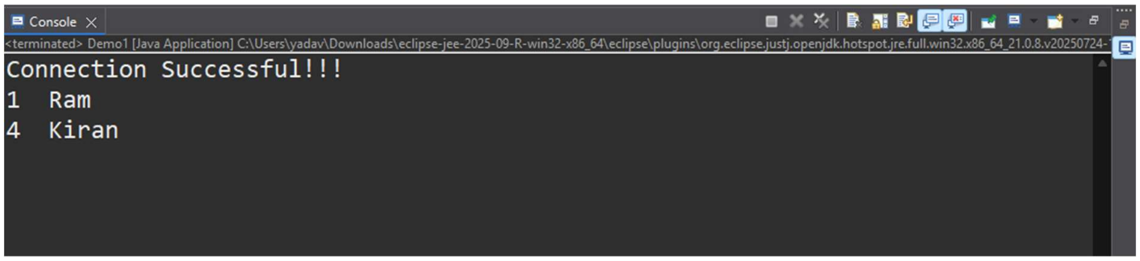
ASSIGNMENT 01 (JDBC)

Q1) ***1. Write a program to select specfic columns (atleast 2 cols) from customers table with multiple condition(atleast two conditions)***

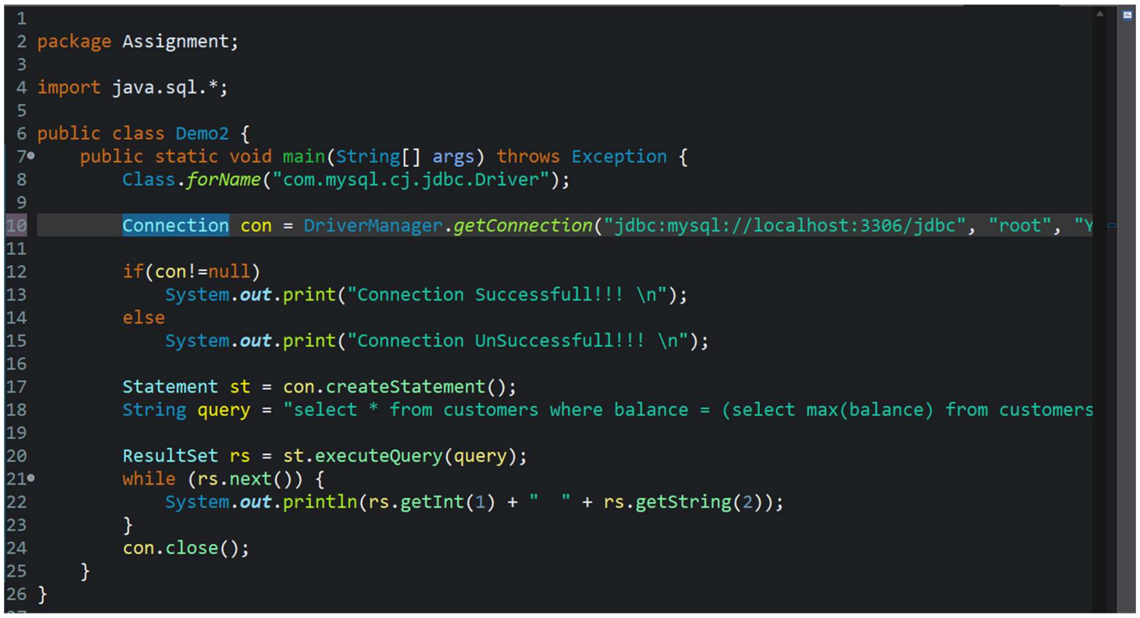


```java
package Assignment;
import java.sql.*;
public class Demo1 {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jdbc",
            "root",
            "Yash@1234"
        );
        if (con != null)
            System.out.println("Connection Successful!!!");
        else
            System.out.println("Connection Unsuccessful!!!");
        Statement st = con.createStatement();
        String query = "SELECT cid, cname FROM customers WHERE city = 'hyd' AND balance > 50000";
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            System.out.println(rs.getInt("cid") + "  " + rs.getString("cname"));
        }

        con.close();
    }
}
```

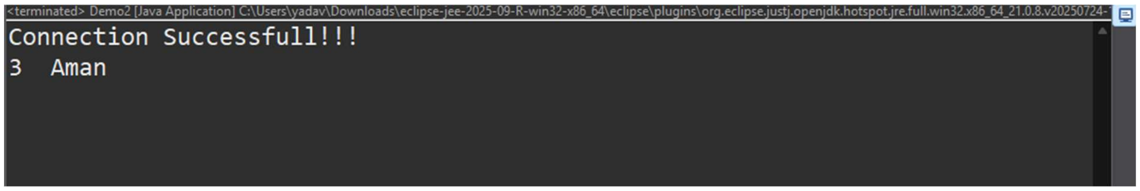


```
<terminated> Demo1 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-
Connection Successful!!!
1   Ram
4   Kiran
```

Q2) ***2. Write a program to demonstrate sub queries. Provide customer details who is having maximum balance using subquery.***

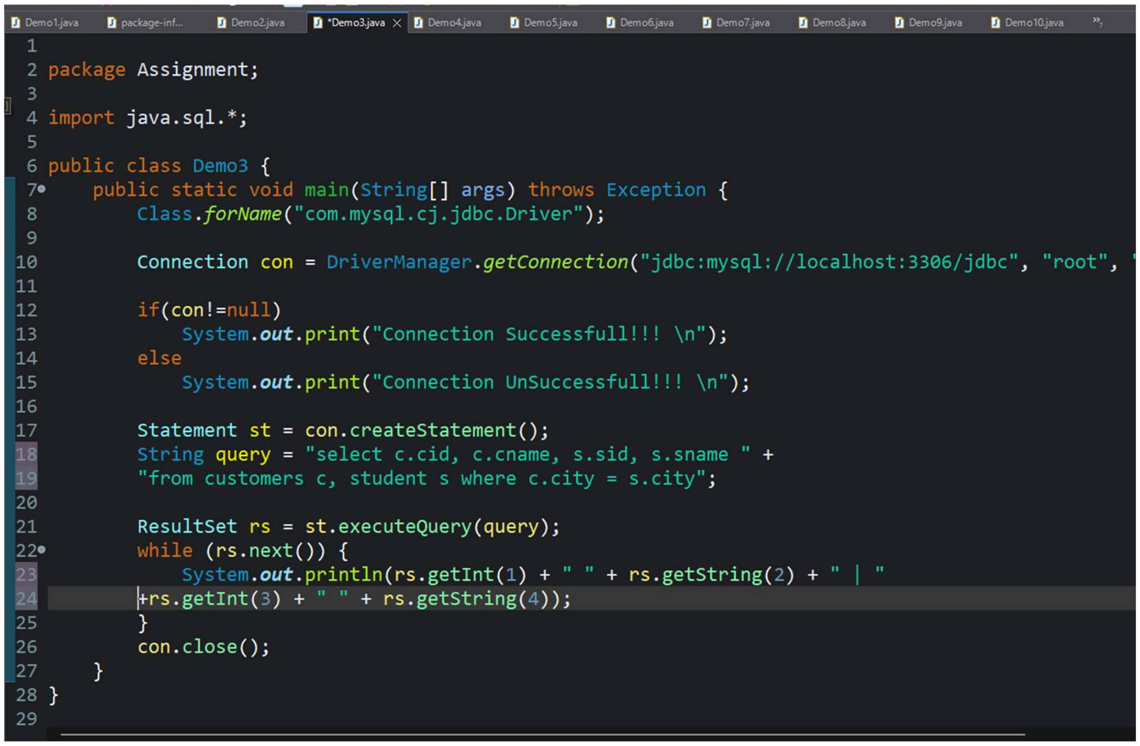


```java

package Assignment;

import java.sql.*;

public class Demo2 {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", "root", "Y

        if(con!=null)
            System.out.print("Connection Successfull!!! \n");
        else
            System.out.print("Connection UnSuccessfull!!! \n");

        Statement st = con.createStatement();
        String query = "select * from customers where balance = (select max(balance) from customers

        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            System.out.println(rs.getInt(1) + "  " + rs.getString(2));
        }
        con.close();
    }
}
```

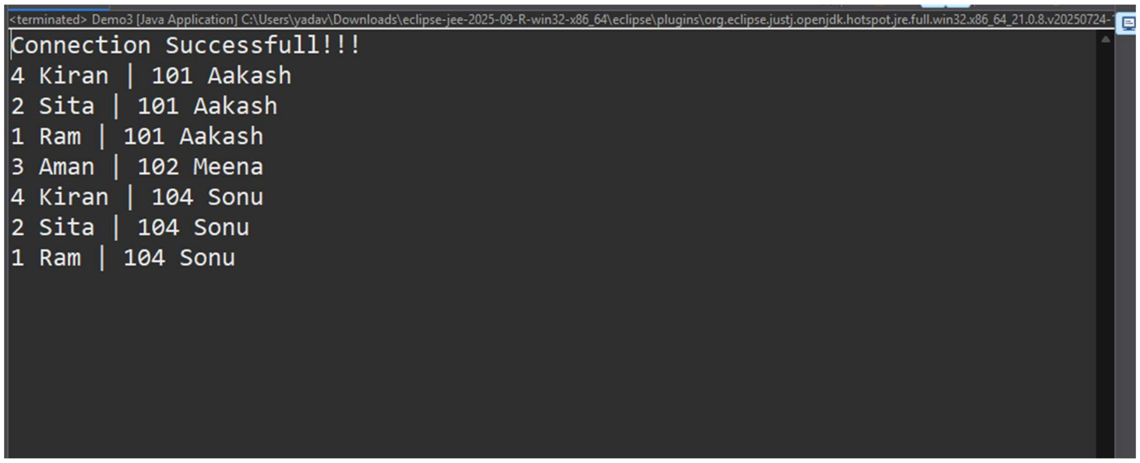


```
<terminated> Demo2 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-
Connection Successfull!!!
3  Aman
```

Q3) ***Write a program to select data from multiple columns of multiple tables. Use customers and student tables to select 2 fields from each table***

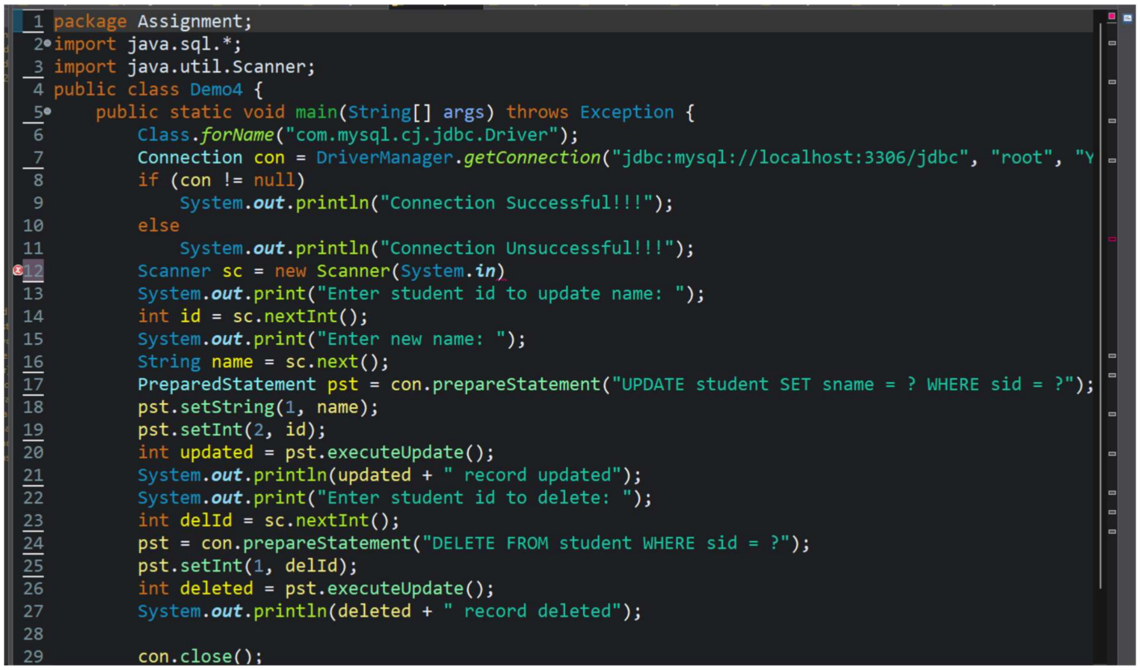


```java

package Assignment;

import java.sql.*;

public class Demo3 {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", "root", '

        if(con!=null)
            System.out.print("Connection Successfull!!! \n");
        else
            System.out.print("Connection UnSuccessfull!!! \n");

        Statement st = con.createStatement();
        String query = "select c.cid, c.cname, s.sid, s.sname " +
        "from customers c, student s where c.city = s.city";

        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            System.out.println(rs.getInt(1) + " " + rs.getString(2) + " | "
        +rs.getInt(3) + " " + rs.getString(4));
        }
        con.close();
    }
}

```

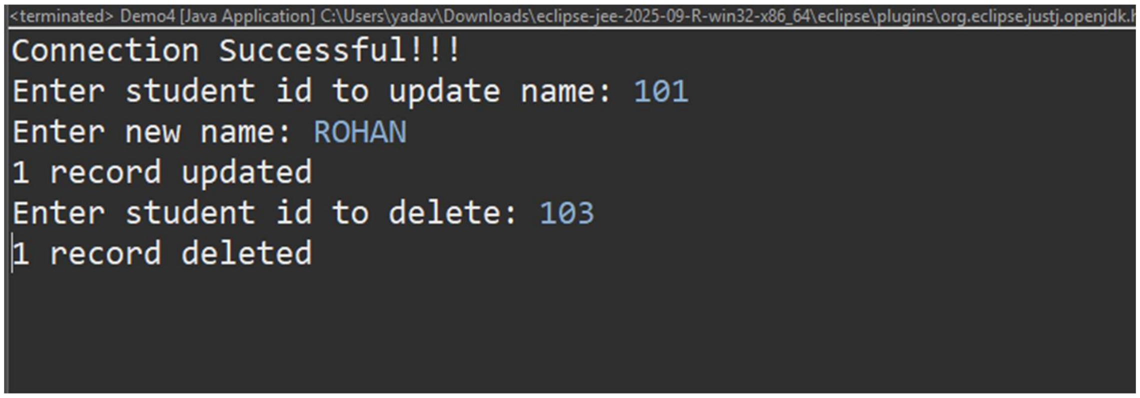


```
<terminated> Demo3 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-
Connection Successfull!!!
4 Kiran | 101 Aakash
2 Sita | 101 Aakash
1 Ram | 101 Aakash
3 Aman | 102 Meena
4 Kiran | 104 Sonu
2 Sita | 104 Sonu
1 Ram | 104 Sonu
```

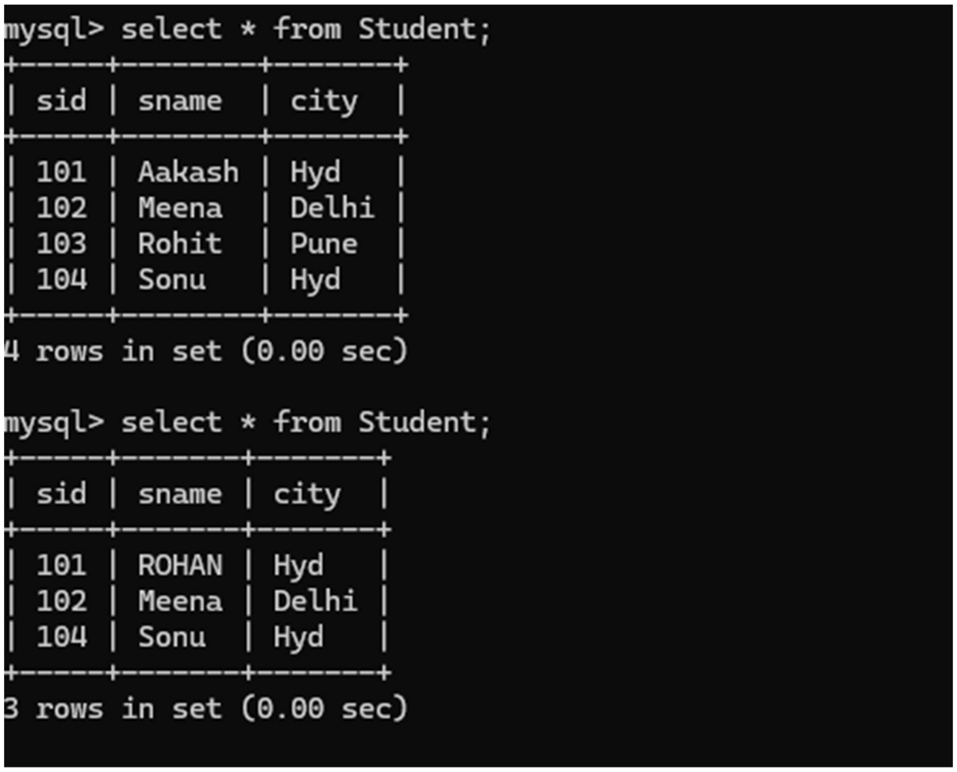
Q4) *Program to delete and update a record of student table based on data read from user*



```
package Assignment;
import java.sql.*;
import java.util.Scanner;
public class Demo4 {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", "root", "Y
        if (con != null)
            System.out.println("Connection Successful!!!");
        else
            System.out.println("Connection Unsuccessful!!!");
        Scanner sc = new Scanner(System.in)
        System.out.print("Enter student id to update name: ");
        int id = sc.nextInt();
        System.out.print("Enter new name: ");
        String name = sc.next();
        PreparedStatement pst = con.prepareStatement("UPDATE student SET sname = ? WHERE sid = ?");
        pst.setString(1, name);
        pst.setInt(2, id);
        int updated = pst.executeUpdate();
        System.out.println(updated + " record updated");
        System.out.print("Enter student id to delete: ");
        int delId = sc.nextInt();
        pst = con.prepareStatement("DELETE FROM student WHERE sid = ?");
        pst.setInt(1, delId);
        int deleted = pst.executeUpdate();
        System.out.println(deleted + " record deleted");

        con.close();
```

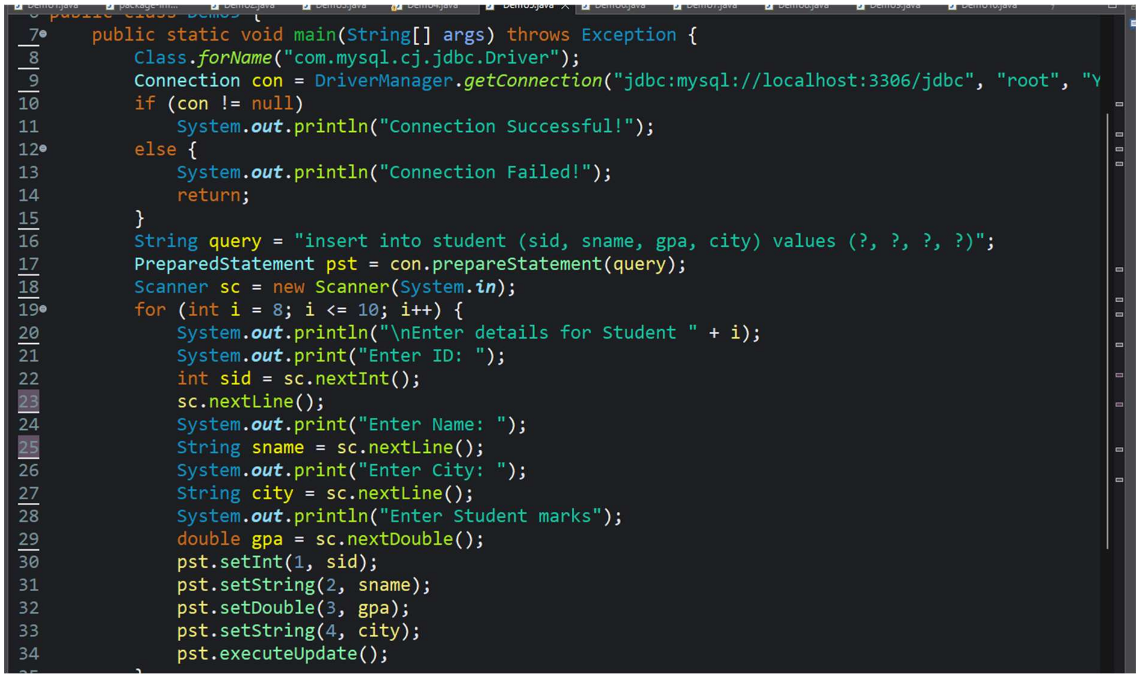


```
<terminated> Demo4 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.
Connection Successful!!!
Enter student id to update name: 101
Enter new name: ROHAN
1 record updated
Enter student id to delete: 103
1 record deleted
```

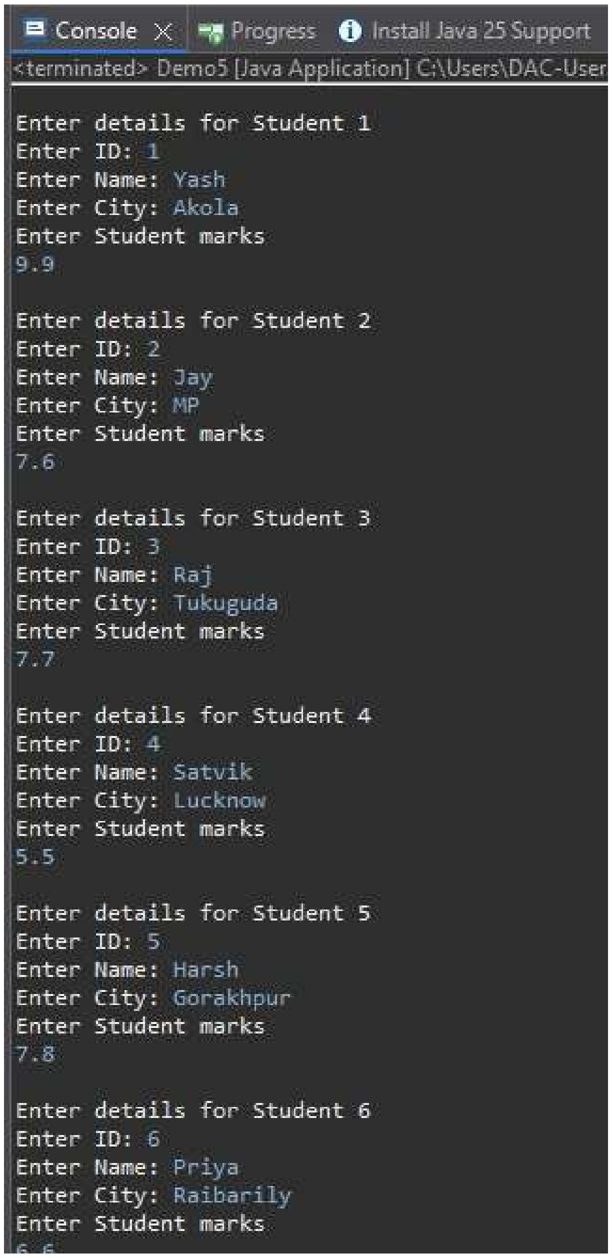


```
mysql> select * from Student;
+-----+--------+-------+
| sid | sname  | city  |
+-----+--------+-------+
| 101 | Aakash | Hyd   |
| 102 | Meena  | Delhi |
| 103 | Rohit  | Pune  |
| 104 | Sonu   | Hyd   |
+-----+--------+-------+
4 rows in set (0.00 sec)

mysql> select * from Student;
+-----+-------+-------+
| sid | sname | city  |
+-----+-------+-------+
| 101 | ROHAN | Hyd   |
| 102 | Meena | Delhi |
| 104 | Sonu  | Hyd   |
+-----+-------+-------+
3 rows in set (0.00 sec)
```

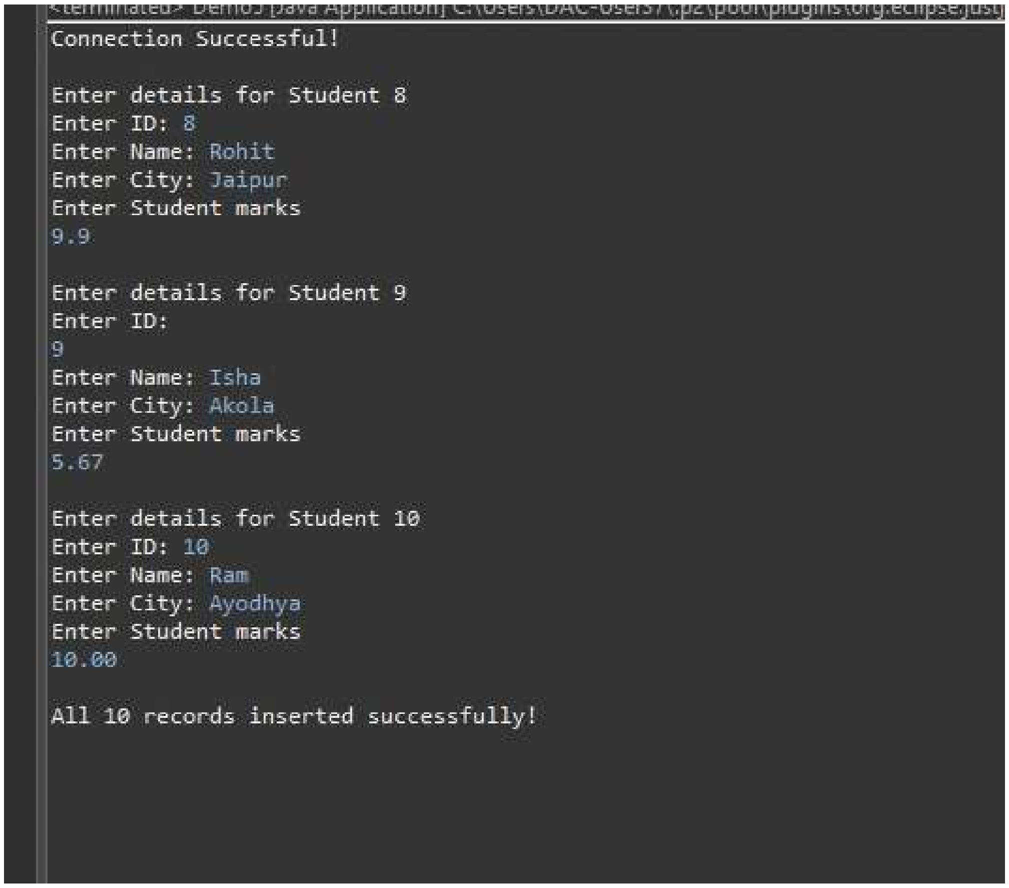
Q5) ***Write program to read multiple student details (10 records) from user and insert them to student table. Use for loop and PreparedStatement***



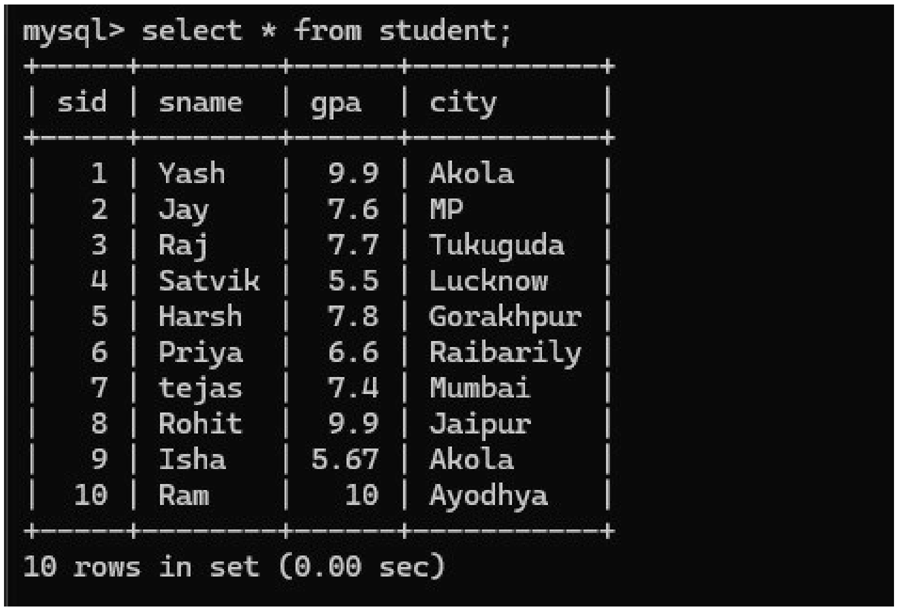
```java
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc", "root", "Y
        if (con != null)
            System.out.println("Connection Successful!");
        else {
            System.out.println("Connection Failed!");
            return;
        }
        String query = "insert into student (sid, sname, gpa, city) values (?, ?, ?, ?)";
        PreparedStatement pst = con.prepareStatement(query);
        Scanner sc = new Scanner(System.in);
        for (int i = 8; i <= 10; i++) {
            System.out.println("\nEnter details for Student " + i);
            System.out.print("Enter ID: ");
            int sid = sc.nextInt();
            sc.nextLine();
            System.out.print("Enter Name: ");
            String sname = sc.nextLine();
            System.out.print("Enter City: ");
            String city = sc.nextLine();
            System.out.println("Enter Student marks");
            double gpa = sc.nextDouble();
            pst.setInt(1, sid);
            pst.setString(2, sname);
            pst.setDouble(3, gpa);
            pst.setString(4, city);
            pst.executeUpdate();
        }
```



```
<terminated> Demo5 [Java Application] C:\Users\DAC-User

Enter details for Student 1
Enter ID: 1
Enter Name: Yash
Enter City: Akola
Enter Student marks
9.9

Enter details for Student 2
Enter ID: 2
Enter Name: Jay
Enter City: MP
Enter Student marks
7.6

Enter details for Student 3
Enter ID: 3
Enter Name: Raj
Enter City: Tukuguda
Enter Student marks
7.7

Enter details for Student 4
Enter ID: 4
Enter Name: Satvik
Enter City: Lucknow
Enter Student marks
5.5

Enter details for Student 5
Enter ID: 5
Enter Name: Harsh
Enter City: Gorakhpur
Enter Student marks
7.8

Enter details for Student 6
Enter ID: 6
Enter Name: Priya
Enter City: Raibarily
Enter Student marks
```

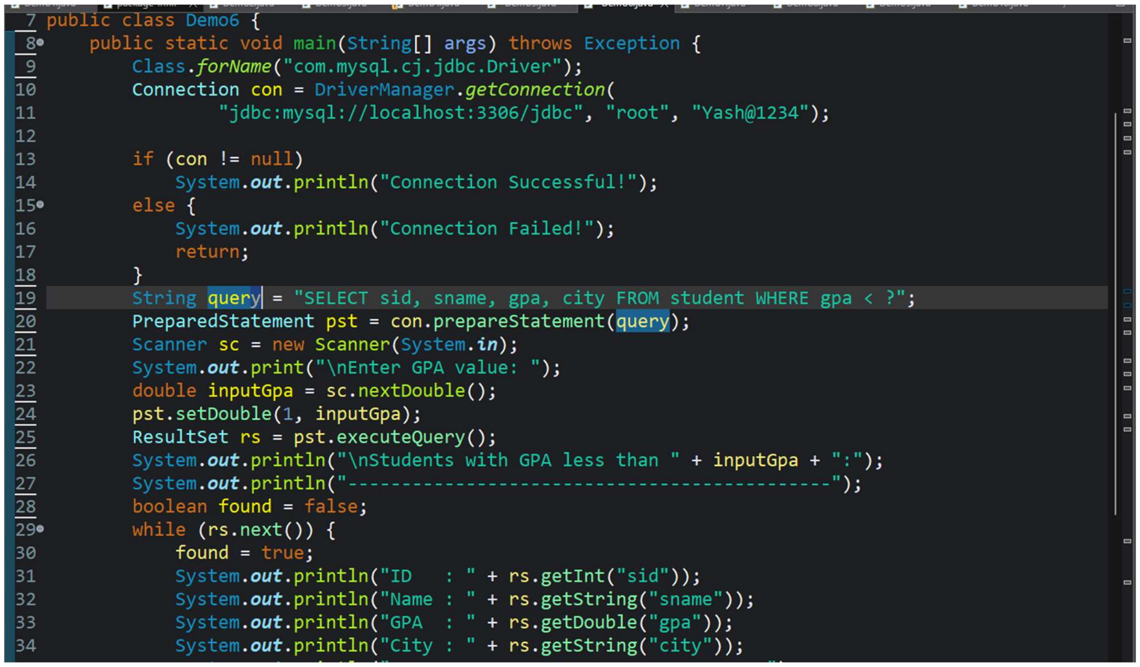


```
<terminated> Demo5 [Java Application] C:\Users\DAC-Users7\.p2\pool\plugins\org.eclipse.just
Connection Successful!

Enter details for Student 8
Enter ID: 8
Enter Name: Rohit
Enter City: Jaipur
Enter Student marks
9.9

Enter details for Student 9
Enter ID:
9
Enter Name: Isha
Enter City: Akola
Enter Student marks
5.67

Enter details for Student 10
Enter ID: 10
Enter Name: Ram
Enter City: Ayodhya
Enter Student marks
10.00

All 10 records inserted successfully!
```

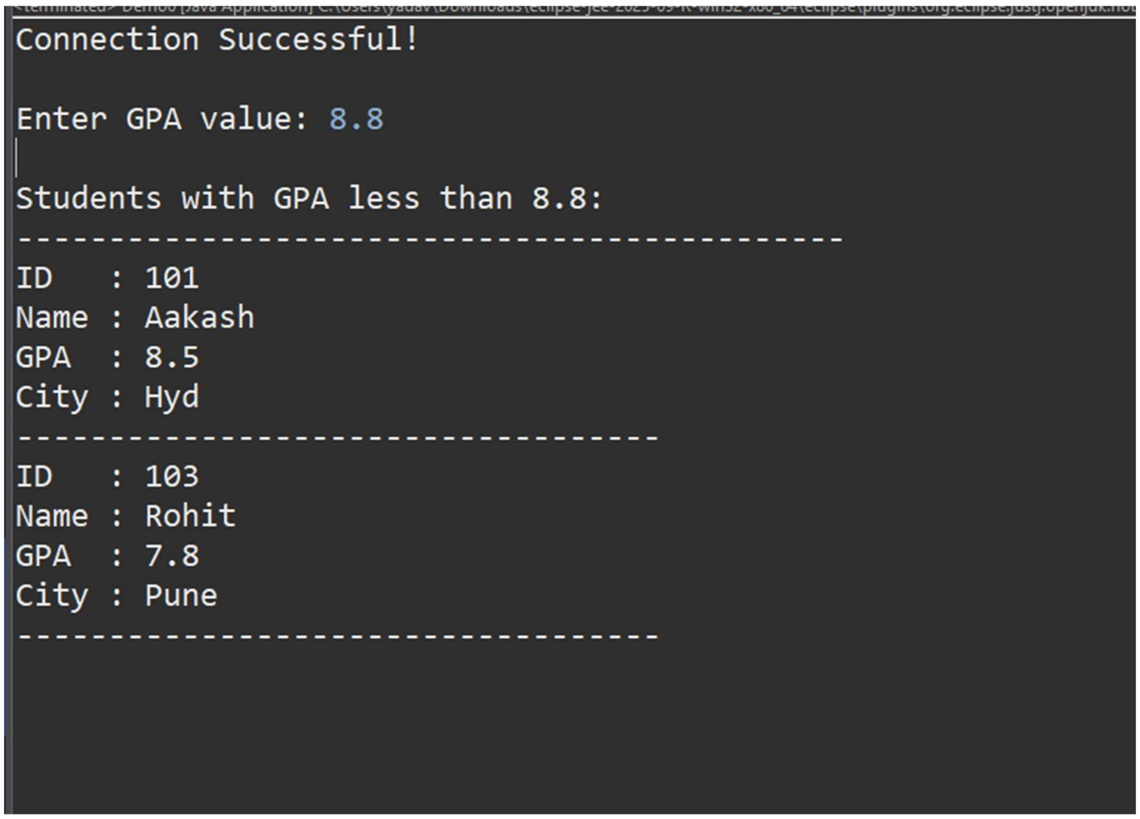


```
mysql> select * from student;
+-----+--------+------+-----------+
| sid | sname  | gpa  | city      |
+-----+--------+------+-----------+
|   1 | Yash   |  9.9 | Akola     |
|   2 | Jay    |  7.6 | MP        |
|   3 | Raj    |  7.7 | Tukuguda  |
|   4 | Satvik |  5.5 | Lucknow   |
|   5 | Harsh  |  7.8 | Gorakhpur |
|   6 | Priya  |  6.6 | Raibarily |
|   7 | tejas  |  7.4 | Mumbai    |
|   8 | Rohit  |  9.9 | Jaipur    |
|   9 | Isha   | 5.67 | Akola     |
|  10 | Ram    |   10 | Ayodhya   |
+-----+--------+------+-----------+
10 rows in set (0.00 sec)
```

Q6) 6. Demonstrate select query using Prepared Statement by reading data from user(Read gpa and provide details of students whose gpa less than that entered)



```java
public class Demo6 {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/jdbc", "root", "Yash@1234");

        if (con != null)
            System.out.println("Connection Successful!");
        else {
            System.out.println("Connection Failed!");
            return;
        }
        String query = "SELECT sid, sname, gpa, city FROM student WHERE gpa < ?";
        PreparedStatement pst = con.prepareStatement(query);
        Scanner sc = new Scanner(System.in);
        System.out.print("\nEnter GPA value: ");
        double inputGpa = sc.nextDouble();
        pst.setDouble(1, inputGpa);
        ResultSet rs = pst.executeQuery();
        System.out.println("\nStudents with GPA less than " + inputGpa + ":");
        System.out.println("---------------------------------------------");
        boolean found = false;
        while (rs.next()) {
            found = true;
            System.out.println("ID   : " + rs.getInt("sid"));
            System.out.println("Name : " + rs.getString("sname"));
            System.out.println("GPA  : " + rs.getDouble("gpa"));
            System.out.println("City : " + rs.getString("city"));
```



```
Connection Successful!

Enter GPA value: 8.8

Students with GPA less than 8.8:
---------------------------------------------
ID   : 101
Name : Aakash
GPA  : 8.5
City : Hyd
-----------------------------------
ID   : 103
Name : Rohit
GPA  : 7.8
City : Pune
-----------------------------------
```

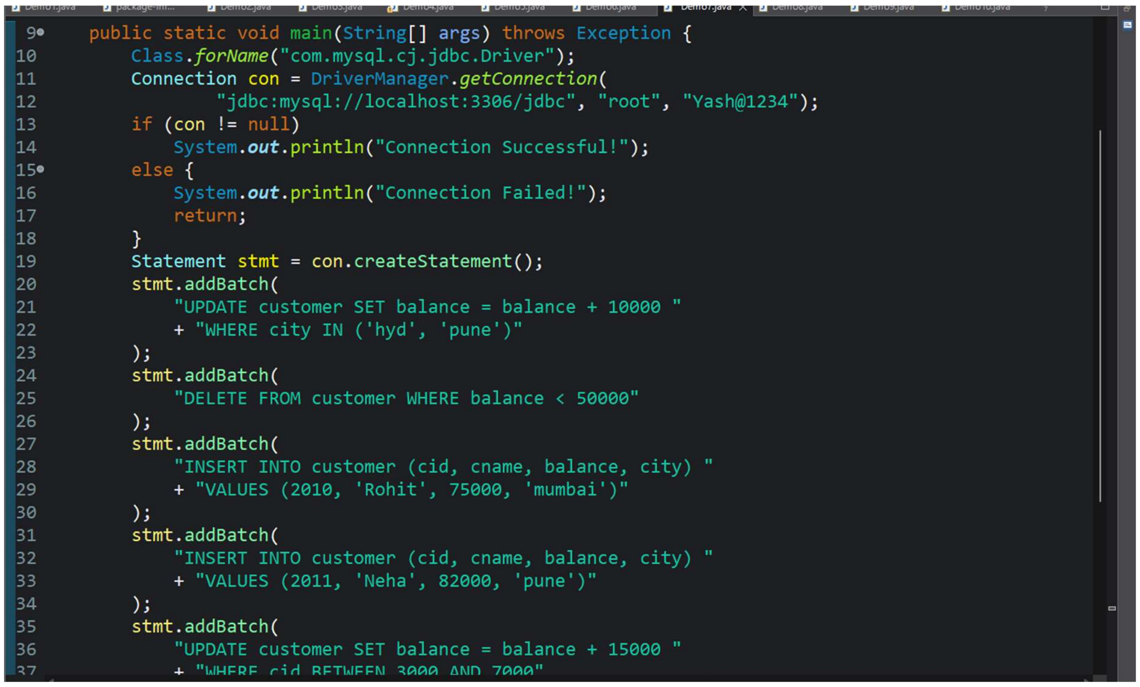
Q7) Write a program to perform jdbc batch processing -

a. update customers balance (add 10000) where city is hyd and pune

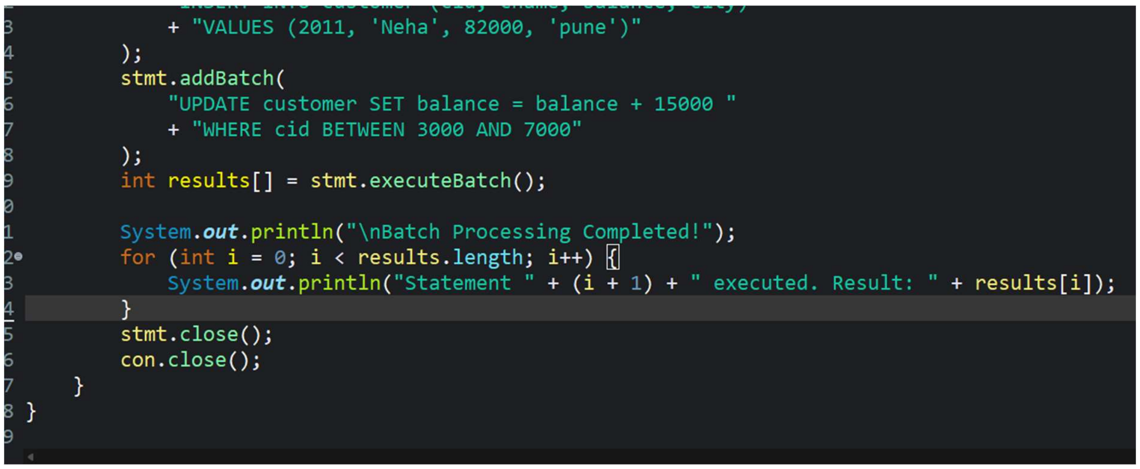
b. Delete customers if balance is less than 50000

c & d. insert 2 customer records -

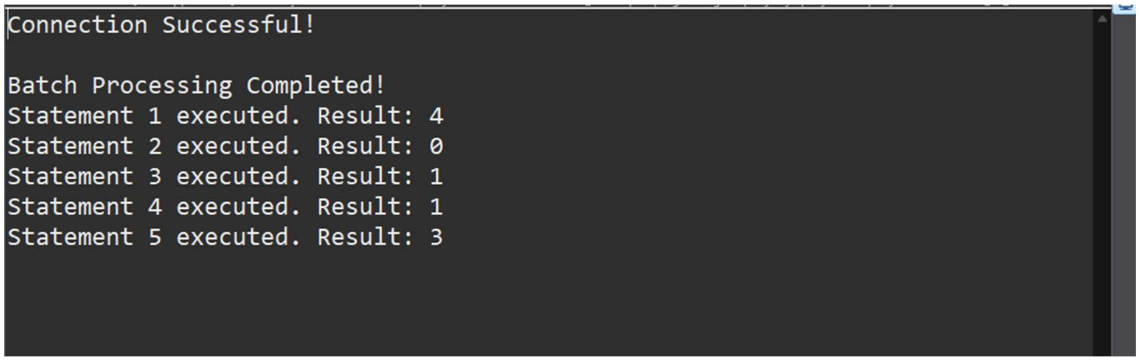
e. update customers with balance with + 15000 if cid between 3000 and 7000



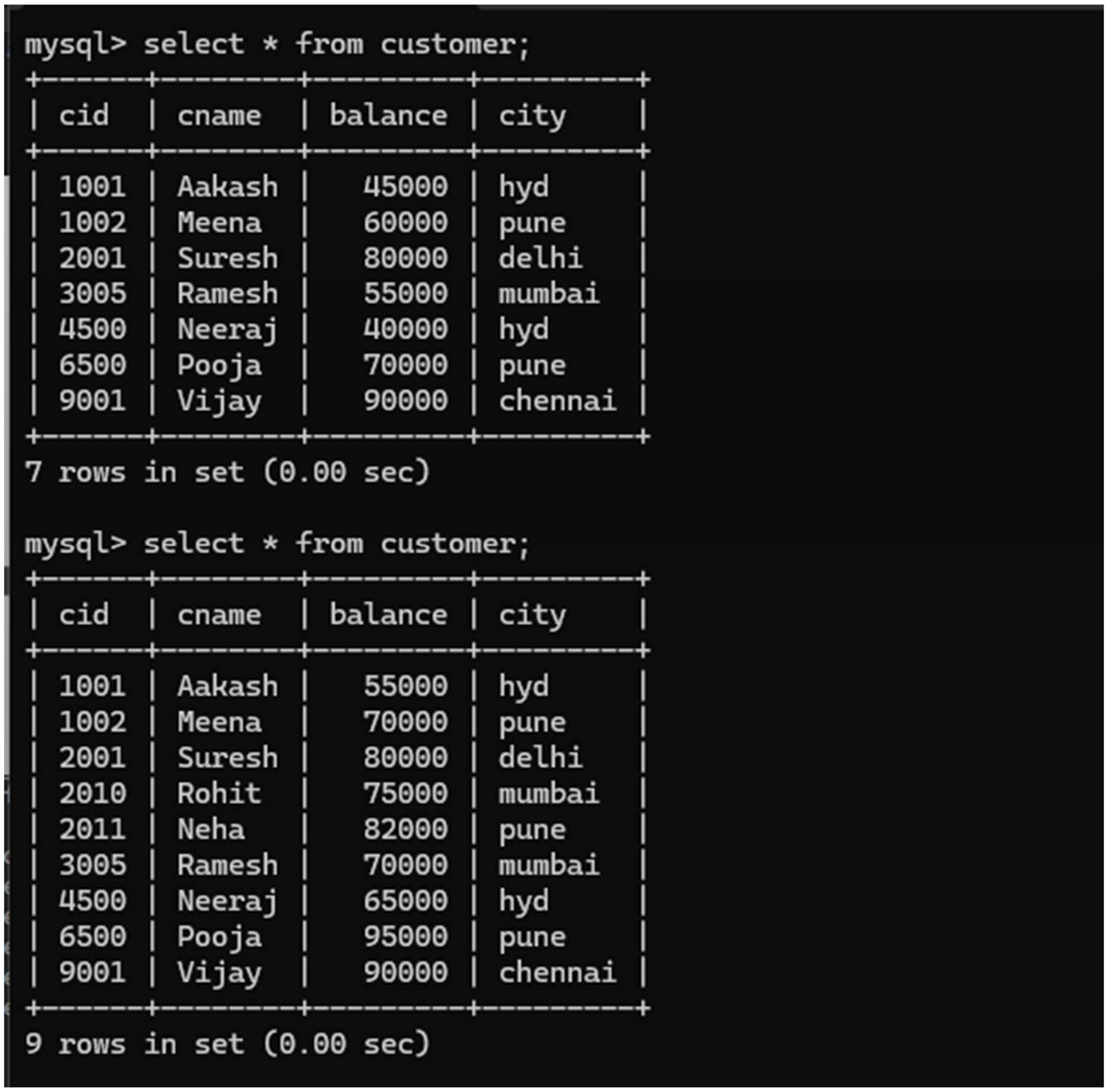
```java
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/jdbc", "root", "Yash@1234");
        if (con != null)
            System.out.println("Connection Successful!");
        else {
            System.out.println("Connection Failed!");
            return;
        }
        Statement stmt = con.createStatement();
        stmt.addBatch(
            "UPDATE customer SET balance = balance + 10000 "
            + "WHERE city IN ('hyd', 'pune')"
        );
        stmt.addBatch(
            "DELETE FROM customer WHERE balance < 50000"
        );
        stmt.addBatch(
            "INSERT INTO customer (cid, cname, balance, city) "
            + "VALUES (2010, 'Rohit', 75000, 'mumbai')"
        );
        stmt.addBatch(
            "INSERT INTO customer (cid, cname, balance, city) "
            + "VALUES (2011, 'Neha', 82000, 'pune')"
        );
        stmt.addBatch(
            "UPDATE customer SET balance = balance + 15000 "
            + "WHERE cid BETWEEN 3000 AND 7000"
```



```java
            "INSERT INTO customer (cid, cname, balance, city)
            + "VALUES (2011, 'Neha', 82000, 'pune')"
        );
        stmt.addBatch(
            "UPDATE customer SET balance = balance + 15000 "
            + "WHERE cid BETWEEN 3000 AND 7000"
        );
        int results[] = stmt.executeBatch();

        System.out.println("\nBatch Processing Completed!");
        for (int i = 0; i < results.length; i++) {
            System.out.println("Statement " + (i + 1) + " executed. Result: " + results[i]);
        }
        stmt.close();
        con.close();
    }
}
```

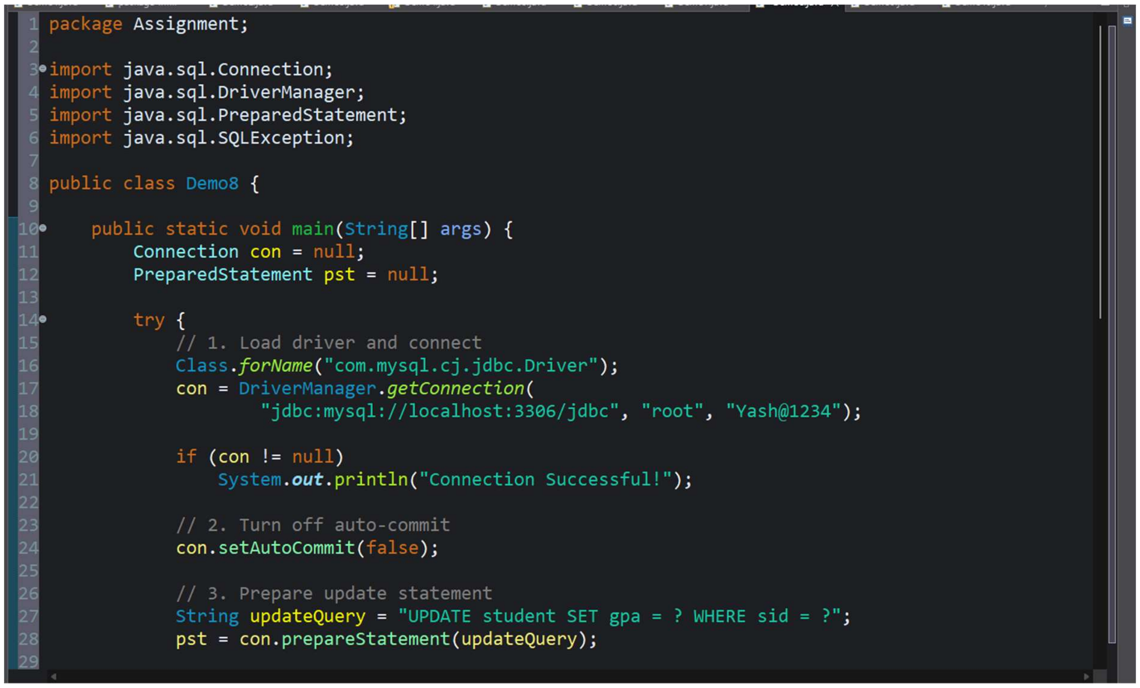


```
Connection Successful!

Batch Processing Completed!
Statement 1 executed. Result: 4
Statement 2 executed. Result: 0
Statement 3 executed. Result: 1
Statement 4 executed. Result: 1
Statement 5 executed. Result: 3
```

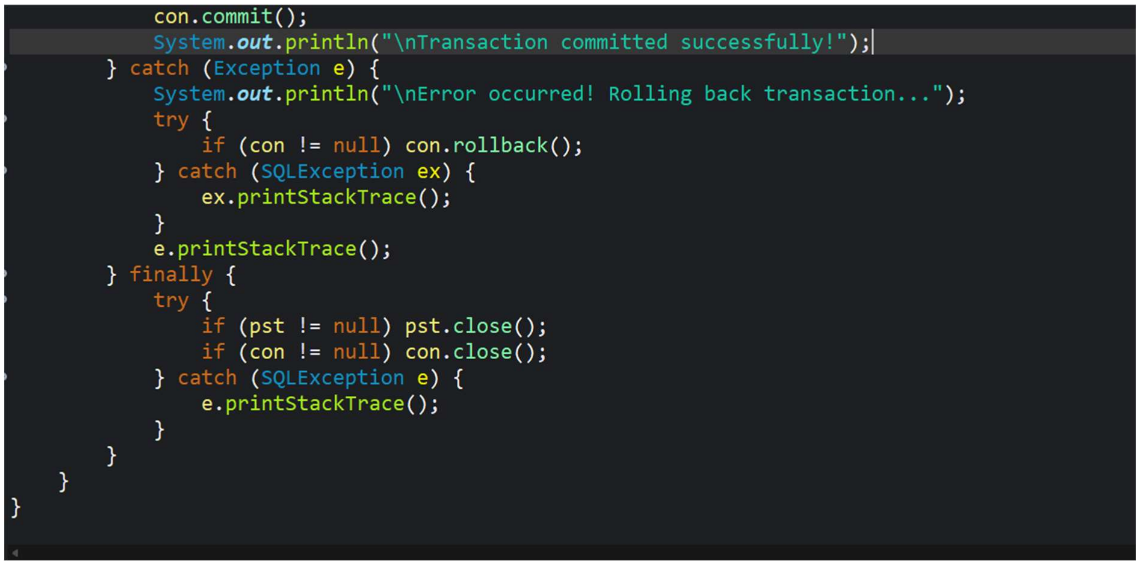


```
mysql> select * from customer;
+------+--------+---------+---------+
| cid  | cname  | balance | city    |
+------+--------+---------+---------+
| 1001 | Aakash |   45000 | hyd     |
| 1002 | Meena  |   60000 | pune    |
| 2001 | Suresh |   80000 | delhi   |
| 3005 | Ramesh |   55000 | mumbai  |
| 4500 | Neeraj |   40000 | hyd     |
| 6500 | Pooja  |   70000 | pune    |
| 9001 | Vijay  |   90000 | chennai |
+------+--------+---------+---------+
7 rows in set (0.00 sec)

mysql> select * from customer;
+------+--------+---------+---------+
| cid  | cname  | balance | city    |
+------+--------+---------+---------+
| 1001 | Aakash |   55000 | hyd     |
| 1002 | Meena  |   70000 | pune    |
| 2001 | Suresh |   80000 | delhi   |
| 2010 | Rohit  |   75000 | mumbai  |
| 2011 | Neha   |   82000 | pune    |
| 3005 | Ramesh |   70000 | mumbai  |
| 4500 | Neeraj |   65000 | hyd     |
| 6500 | Pooja  |   95000 | pune    |
| 9001 | Vijay  |   90000 | chennai |
+------+--------+---------+---------+
9 rows in set (0.00 sec)
```

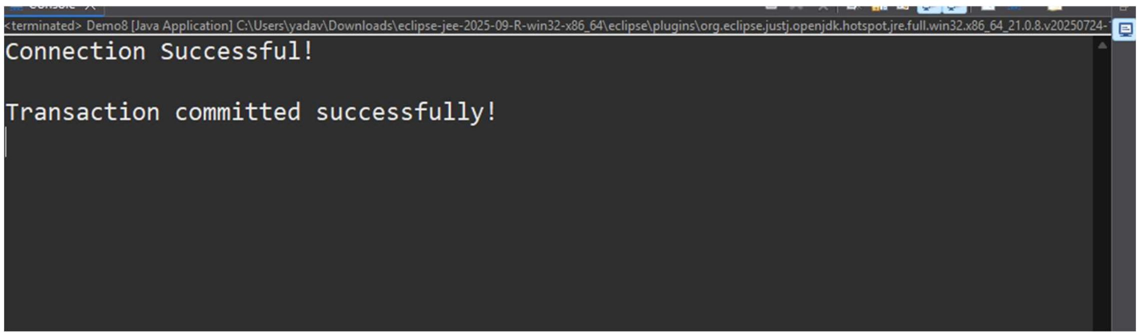
Q8) 8. Demonstrate transaction management - with student table (take suitable case study)



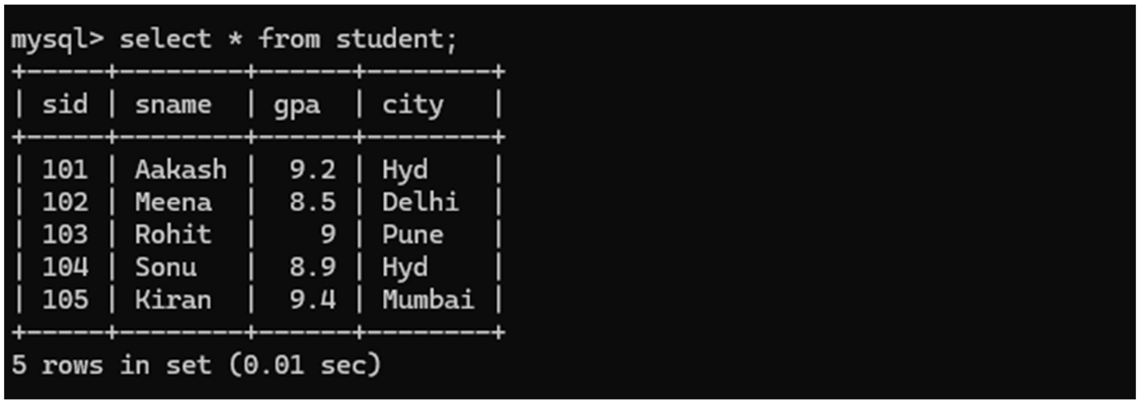
```java
package Assignment;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class Demo8 {

    public static void main(String[] args) {
        Connection con = null;
        PreparedStatement pst = null;

        try {
            // 1. Load driver and connect
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/jdbc", "root", "Yash@1234");

            if (con != null)
                System.out.println("Connection Successful!");

            // 2. Turn off auto-commit
            con.setAutoCommit(false);

            // 3. Prepare update statement
            String updateQuery = "UPDATE student SET gpa = ? WHERE sid = ?";
            pst = con.prepareStatement(updateQuery);
```



```java
            con.commit();
            System.out.println("\nTransaction committed successfully!");
        } catch (Exception e) {
            System.out.println("\nError occurred! Rolling back transaction...");
            try {
                if (con != null) con.rollback();
            } catch (SQLException ex) {
                ex.printStackTrace();
            }
            e.printStackTrace();
        } finally {
            try {
                if (pst != null) pst.close();
                if (con != null) con.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

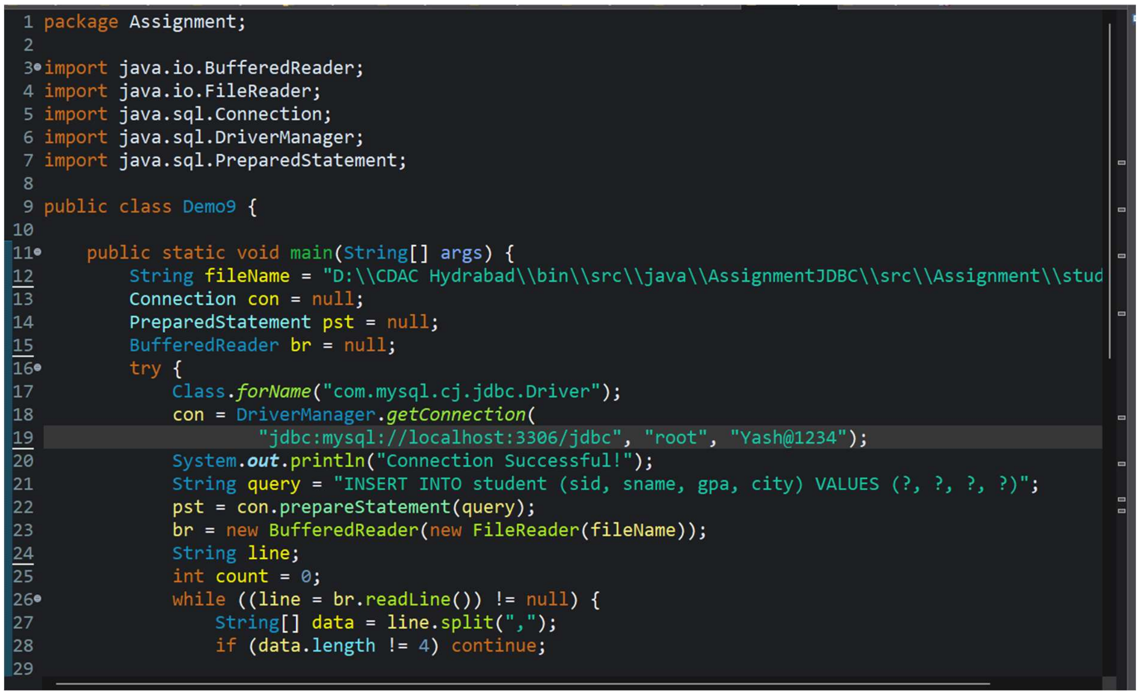


```
<terminated> Demo8 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-
Connection Successful!

Transaction committed successfully!
```

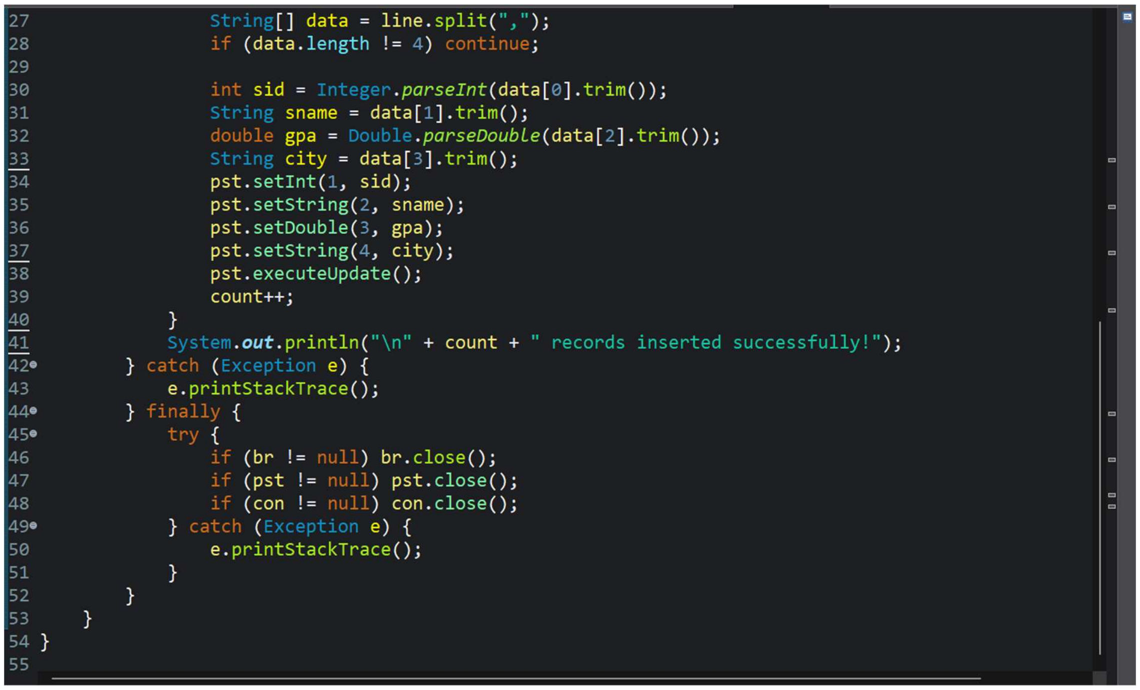


```
mysql> select * from student;
+-----+--------+-----+--------+
| sid | sname  | gpa | city   |
+-----+--------+-----+--------+
| 101 | Aakash | 9.2 | Hyd    |
| 102 | Meena  | 8.5 | Delhi  |
| 103 | Rohit  |   9 | Pune   |
| 104 | Sonu   | 8.9 | Hyd    |
| 105 | Kiran  | 9.4 | Mumbai |
+-----+--------+-----+--------+
5 rows in set (0.01 sec)
```

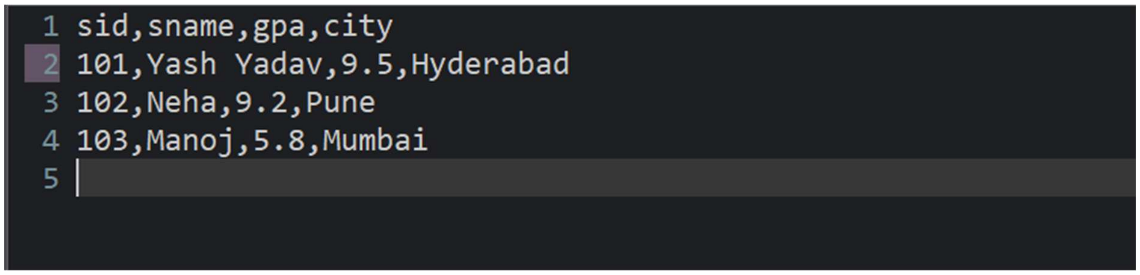
Q9) 9. program to Read data from a text file (student.txt) and insert into student table(Use prepared statement)



```java
package Assignment;

import java.io.BufferedReader;
import java.io.FileReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class Demo9 {

    public static void main(String[] args) {
        String fileName = "D:\\CDAC Hydrabad\\bin\\src\\java\\AssignmentJDBC\\src\\Assignment\\stud
        Connection con = null;
        PreparedStatement pst = null;
        BufferedReader br = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/jdbc", "root", "Yash@1234");
            System.out.println("Connection Successful!");
            String query = "INSERT INTO student (sid, sname, gpa, city) VALUES (?, ?, ?, ?)";
            pst = con.prepareStatement(query);
            br = new BufferedReader(new FileReader(fileName));
            String line;
            int count = 0;
            while ((line = br.readLine()) != null) {
                String[] data = line.split(",");
                if (data.length != 4) continue;

```



```
                String[] data = line.split(",");
                if (data.length != 4) continue;

                int sid = Integer.parseInt(data[0].trim());
                String sname = data[1].trim();
                double gpa = Double.parseDouble(data[2].trim());
                String city = data[3].trim();
                pst.setInt(1, sid);
                pst.setString(2, sname);
                pst.setDouble(3, gpa);
                pst.setString(4, city);
                pst.executeUpdate();
                count++;
            }
            System.out.println("\n" + count + " records inserted successfully!");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (br != null) br.close();
                if (pst != null) pst.close();
                if (con != null) con.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```



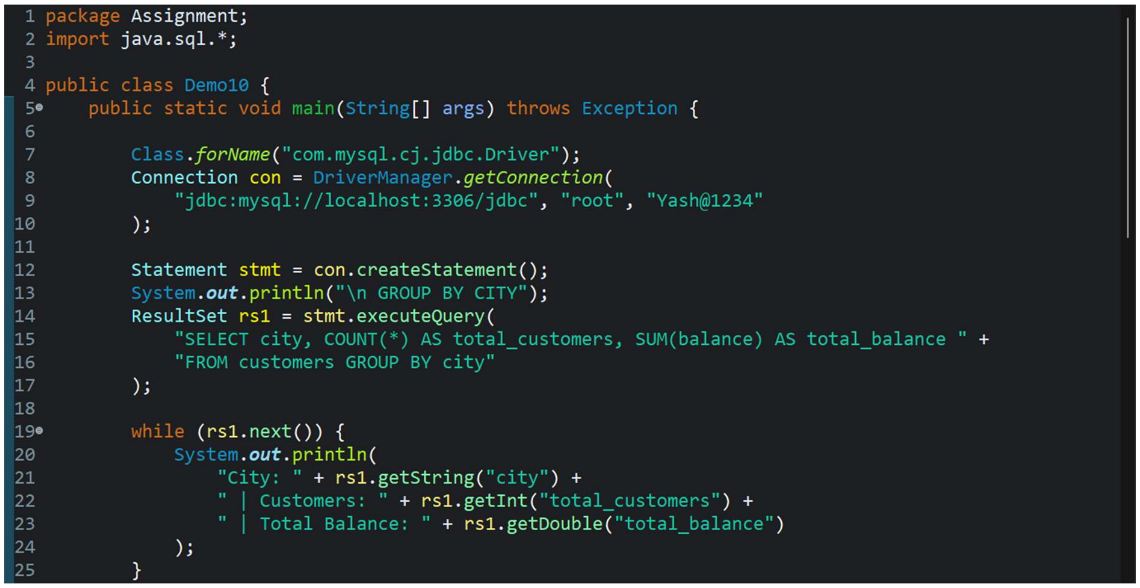
```
sid,sname,gpa,city
101,Yash Yadav,9.5,Hyderabad
102,Neha,9.2,Pune
103,Manoj,5.8,Mumbai

```

Q10)Program to demonstrate where, group by, order by, having with customers table

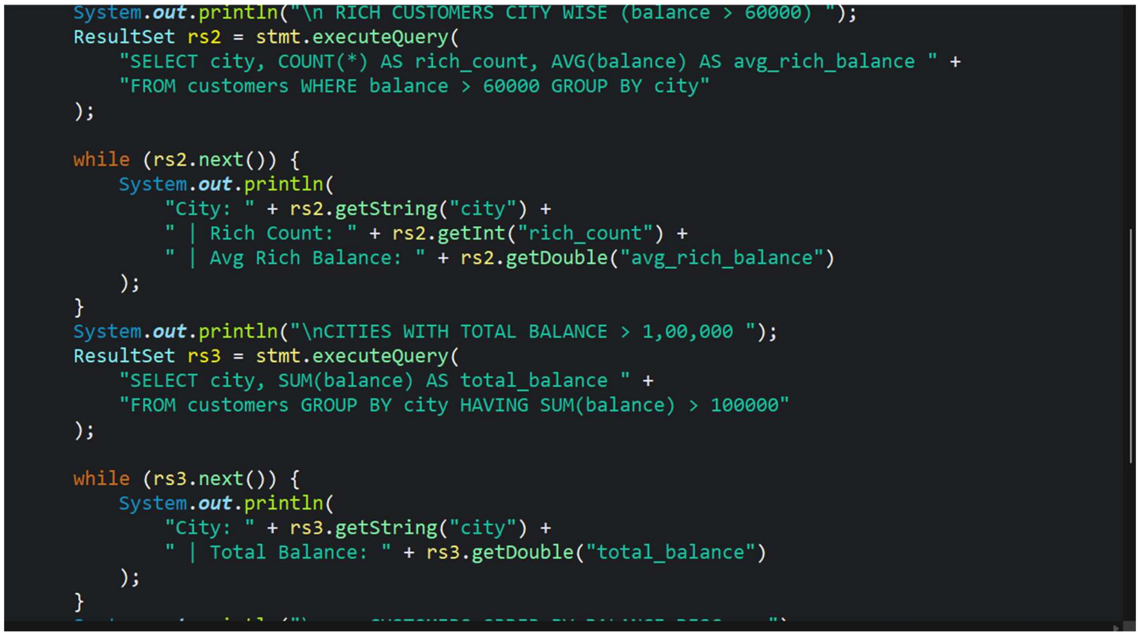
(find customers group by city and provide details)

(find rich customer group by city)

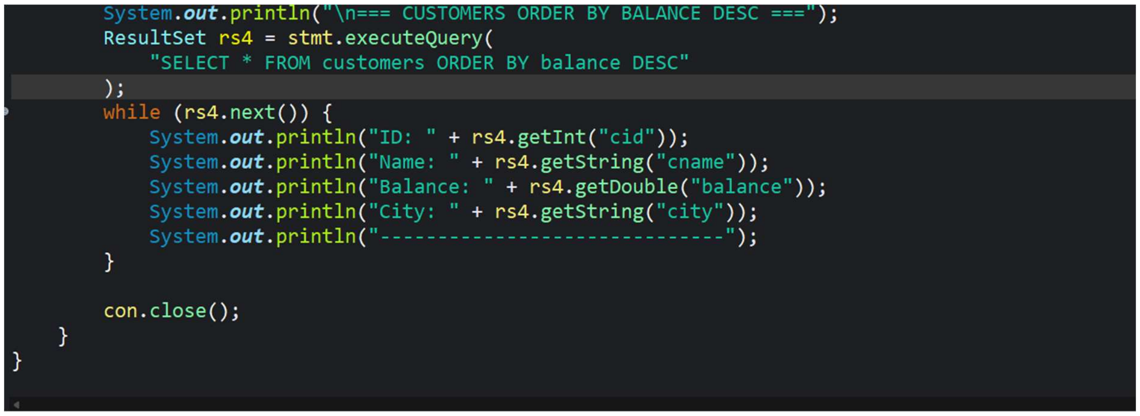
(Use aggregate functions)



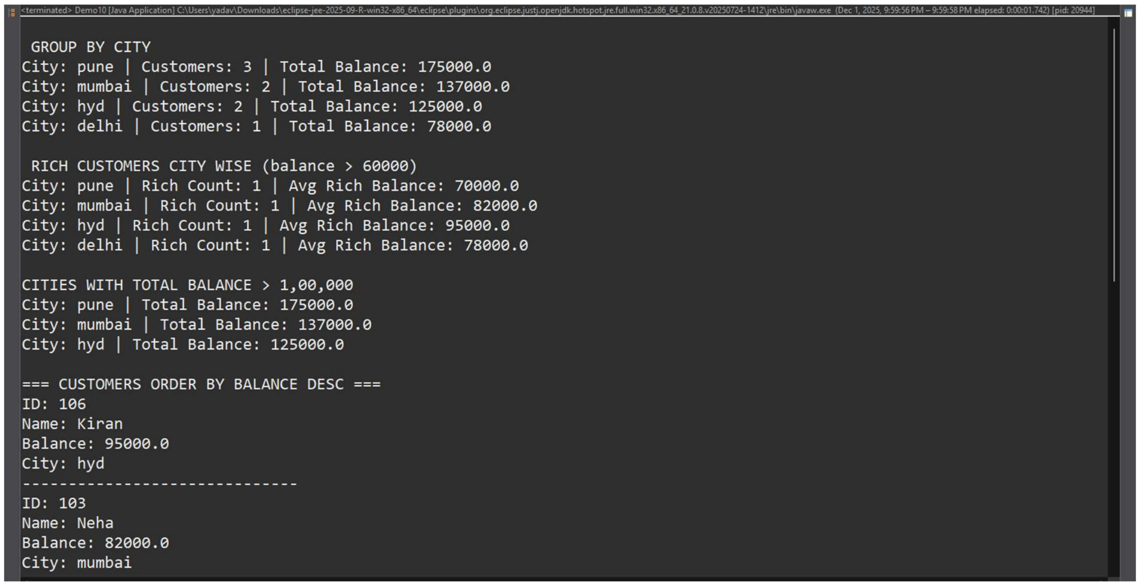
```java
package Assignment;
import java.sql.*;

public class Demo10 {
    public static void main(String[] args) throws Exception {

        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jdbc", "root", "Yash@1234"
        );

        Statement stmt = con.createStatement();
        System.out.println("\n GROUP BY CITY");
        ResultSet rs1 = stmt.executeQuery(
            "SELECT city, COUNT(*) AS total_customers, SUM(balance) AS total_balance " +
            "FROM customers GROUP BY city"
        );

        while (rs1.next()) {
            System.out.println(
                "City: " + rs1.getString("city") +
                " | Customers: " + rs1.getInt("total_customers") +
                " | Total Balance: " + rs1.getDouble("total_balance")
            );
        }
```



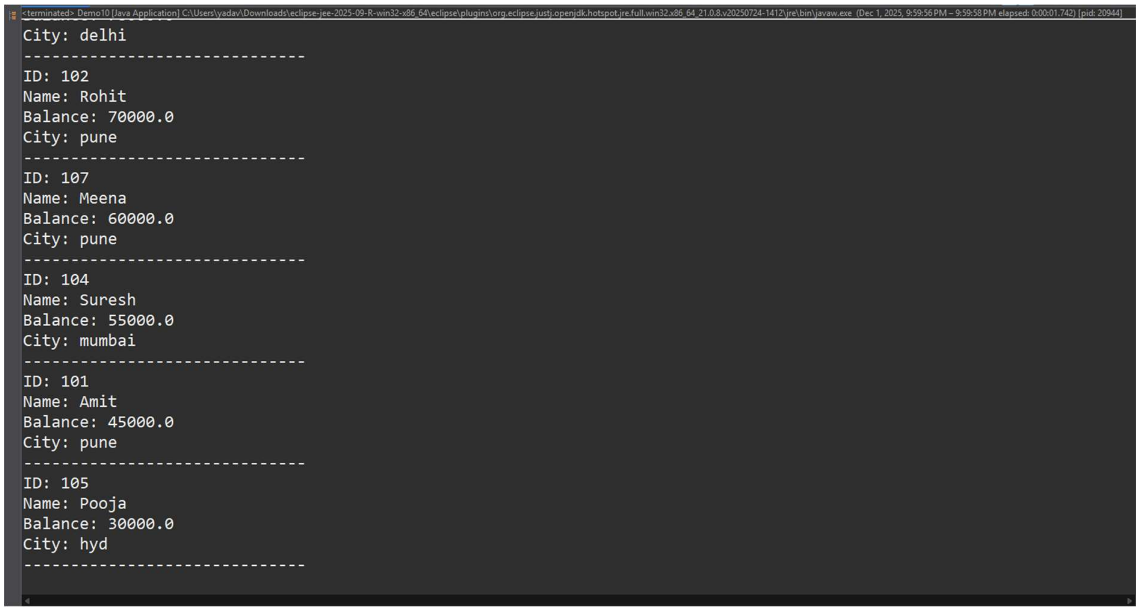
```java
        System.out.println("\n RICH CUSTOMERS CITY WISE (balance > 60000) ");
        ResultSet rs2 = stmt.executeQuery(
            "SELECT city, COUNT(*) AS rich_count, AVG(balance) AS avg_rich_balance " +
            "FROM customers WHERE balance > 60000 GROUP BY city"
        );

        while (rs2.next()) {
            System.out.println(
                "City: " + rs2.getString("city") +
                " | Rich Count: " + rs2.getInt("rich_count") +
                " | Avg Rich Balance: " + rs2.getDouble("avg_rich_balance")
            );
        }
        System.out.println("\nCITIES WITH TOTAL BALANCE > 1,00,000 ");
        ResultSet rs3 = stmt.executeQuery(
            "SELECT city, SUM(balance) AS total_balance " +
            "FROM customers GROUP BY city HAVING SUM(balance) > 100000"
        );

        while (rs3.next()) {
            System.out.println(
                "City: " + rs3.getString("city") +
                " | Total Balance: " + rs3.getDouble("total_balance")
            );
        }
```



```java
        System.out.println("\n=== CUSTOMERS ORDER BY BALANCE DESC ===");
        ResultSet rs4 = stmt.executeQuery(
            "SELECT * FROM customers ORDER BY balance DESC"
        );
        while (rs4.next()) {
            System.out.println("ID: " + rs4.getInt("cid"));
            System.out.println("Name: " + rs4.getString("cname"));
            System.out.println("Balance: " + rs4.getDouble("balance"));
            System.out.println("City: " + rs4.getString("city"));
            System.out.println("-------------------------------");
        }

        con.close();
    }
}
```



```
<terminated> Demo10 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe  (Dec 1, 2025, 9:59:56 PM – 9:59:58 PM elapsed: 0:00:01.742) [pid: 20944]

 GROUP BY CITY
City: pune | Customers: 3 | Total Balance: 175000.0
City: mumbai | Customers: 2 | Total Balance: 137000.0
City: hyd | Customers: 2 | Total Balance: 125000.0
City: delhi | Customers: 1 | Total Balance: 78000.0

 RICH CUSTOMERS CITY WISE (balance > 60000)
City: pune | Rich Count: 1 | Avg Rich Balance: 70000.0
City: mumbai | Rich Count: 1 | Avg Rich Balance: 82000.0
City: hyd | Rich Count: 1 | Avg Rich Balance: 95000.0
City: delhi | Rich Count: 1 | Avg Rich Balance: 78000.0

CITIES WITH TOTAL BALANCE > 1,00,000
City: pune | Total Balance: 175000.0
City: mumbai | Total Balance: 137000.0
City: hyd | Total Balance: 125000.0

=== CUSTOMERS ORDER BY BALANCE DESC ===
ID: 106
Name: Kiran
Balance: 95000.0
City: hyd
------------------------------
ID: 103
Name: Neha
Balance: 82000.0
City: mumbai
```



```
<terminated> Demo10 [Java Application] C:\Users\yadav\Downloads\eclipse-jee-2025-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe  (Dec 1, 2025, 9:59:56 PM – 9:59:58 PM elapsed: 0:00:01.742) [pid: 20944]
City: delhi
------------------------------
ID: 102
Name: Rohit
Balance: 70000.0
City: pune
------------------------------
ID: 107
Name: Meena
Balance: 60000.0
City: pune
------------------------------
ID: 104
Name: Suresh
Balance: 55000.0
City: mumbai
------------------------------
ID: 101
Name: Amit
Balance: 45000.0
City: pune
------------------------------
ID: 105
Name: Pooja
Balance: 30000.0
City: hyd
------------------------------
```