Practical-3

Aim:

3.6 Consider Bank Table with attributes AccountNo, Customer Name, Balance, Phone and Address. Write a JDBC Program which allows insertion, updation and deletion of record in Bank Table. Print values of all customers whose balance is greater then specified amount. (Hint: program should be menu driven).

Description:

ResultSet:

The SQL statements that read data from a database query, return the data in a result set. The SELECT statement is the standard way to select rows from a database and view them in a result set. The *java.sql.ResultSet* interface represents the result set of a database query.

A ResultSet object maintains a cursor that points to the current row in the result set. The term "result set" refers to the row and column data contained in a ResultSet object.

The methods of the ResultSet interface can be broken down into three categories –

- Navigational methods: Used to move the cursor around.
- **Get methods:** Used to view the data in the columns of the current row being pointed by the cursor.
- **Update methods:** Used to update the data in the columns of the current row. The updates can then be updated in the underlying database as well.

The cursor is movable based on the properties of the ResultSet. These properties are designated when the corresponding Statement that generates the ResultSet is created.

JDBC provides the following connection methods to create statements with desired ResultSet

- createStatement(int RSType, int RSConcurrency);
- prepareStatement(String SQL, int RSType, int RSConcurrency);
- prepareCall(String sql, int RSType, int RSConcurrency);

The first argument indicates the type of a ResultSet object and the second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable.

Type of ResultSet

The possible RSType are given below. If you do not specify any ResultSet type, you will automatically get one that is TYPE_FORWARD_ONLY.

Туре	Description
ResultSet.TYPE_FORWARD_ONLY	The cursor can only move forward in the result set.
ResultSet.TYPE_SCROLL_INSENSITIVE	The cursor can scroll forward and backward, and the result set is not sensitive to changes made by others to the database that occur after the result set was created.
ResultSet.TYPE_SCROLL_SENSITIVE.	The cursor can scroll forward and backward, and the result set is sensitive to changes made by others to the database that occur after the result set was created.

Program code:

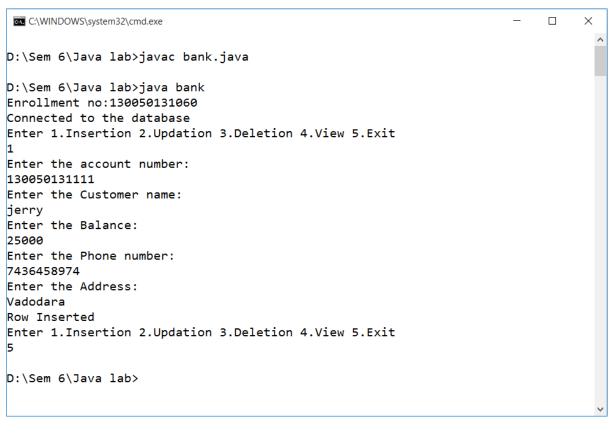
Bank.java

```
import java.util.*;
import java.sql.*;
class bank{
public static void main(String args[]){
int n;
long Account No;
long Phone No;
int Balance;
String Name;
String Address;
Scanner sc=new Scanner(System.in);
ConnectionDb cdb=new ConnectionDb();
Connection con=cdb.connect("localhost", "s2b130050131060");
Statement stmt=con.createStatement();
System.out.println("Connected to the database");
System.out.println("Enter 1.Insertion 2.Updation 3.Deletion 4.View
5.Exit");
```

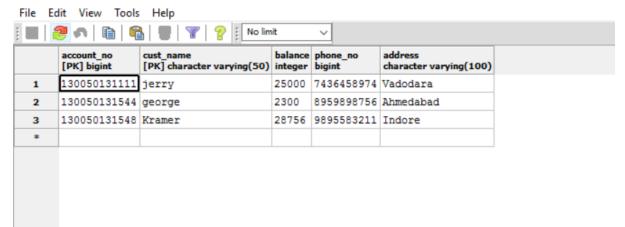
```
n=sc.nextInt();
switch(n) {
case 1:
//Insertion
try{
System.out.println("Enter the account number:");
Account No=sc.nextLong();
System.out.println("Enter the Customer name:");
Name=sc.next();
System.out.println("Enter the Balance:");
Balance=sc.nextInt();
System.out.println("Enter the Phone number:");
Phone No=sc.nextLong();
System.out.println("Enter the Address:");
Address=sc.next();
stmt.executeUpdate("Insert into bank
values("+Account No+",'"+Name+"',"+Balance+","+Phone No+",'"+Address+"');")
System.out.println("Row Inserted");
catch(SQLException se){
System.out.println("Account exist");
break;
case 2:
//Updation
try{
System.out.println("Enter the account number:");
Account No=sc.nextLong();
System.out.println("Enter the Customer name:");
Name=sc.next();
System.out.println("Enter the Balance:");
Balance=sc.nextInt();
System.out.println("Enter the Phone number:");
Phone No=sc.nextLong();
System.out.println("Enter the Address:");
Address=sc.next();
stmt.executeUpdate("update bank set
Cust Name='"+Name+"', Balance="+Balance+", Phone No="+Phone No+", Address='"+A
ddress+"' where Account No="+Account No+";");
System.out.println("Row Updated");
}
catch(SQLException se){
System.out.println(se);
1
break:
case 3:
//Deletion
System.out.println("Enter the account number:");
Account No=sc.nextLong();
stmt.executeUpdate("delete from bank where Account No="+Account No+";");
```

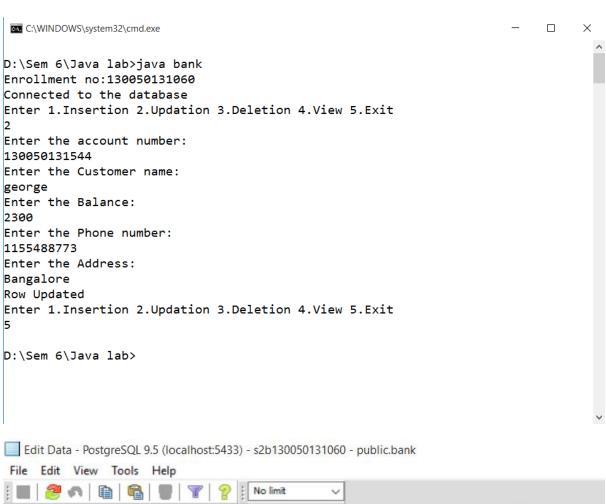
```
System.out.println("Row Deleted");
catch(SQLException se){
System.out.println("Account number doesnot exist");
break;
case 4:
//view
try{
System.out.println("View Balance greater than:");
Balance=sc.nextInt();
ResultSet rs=stmt.executeQuery("Select * from bank where Balance
>="+Balance+";");
while(rs.next()){
Account No=rs.getLong("Account No");
Name=rs.getString("Cust_Name");
Balance=rs.getInt("Balance");
Phone No=rs.getLong("Phone No");
Address=rs.getString("Address");
System.out.print("Account no:"+Account No);
System.out.print("\tCust_name:"+Name);
System.out.print("\tBalance:"+Balance);
System.out.print("\tPhone no:"+Phone No);
System.out.print("\tAddress:"+Address);
System.out.println();
catch(SQLException se){
System.out.println(se);
}
break;
}while (n<=4);</pre>
catch(SQLException se){
System.out.println(se);
}
}
}
```

Output:

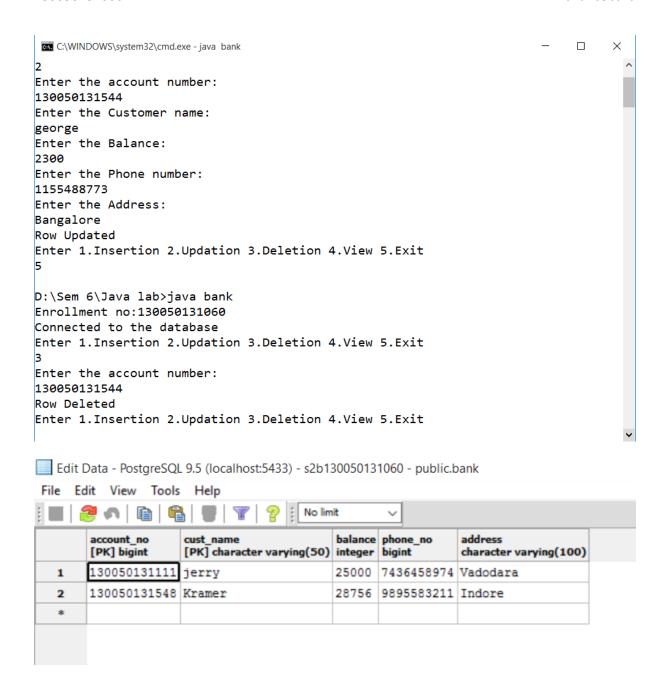


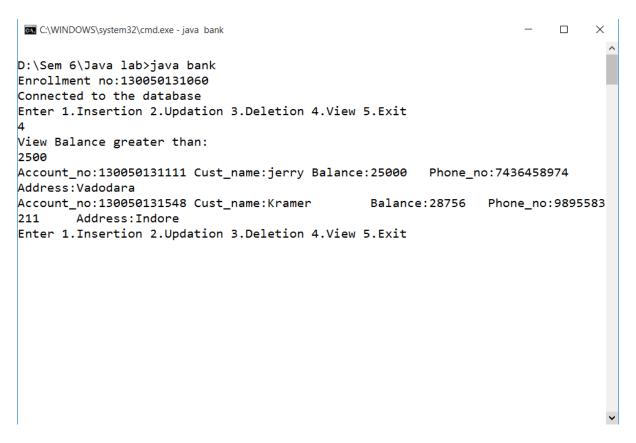
Edit Data - PostgreSQL 9.5 (localhost:5433) - s2b130050131060 - public.bank





Sa	account_no [PK] bigint	cust_name [PK] character varying(50)		phone_no bigint	address character varying(100)
	130050131111	jerry	25000	7436458974	Vadodara
	130050131544	george	2300	1155488773	Bangalore
	130050131548	Kramer	28756	9895583211	Indore





Edit Data - PostgreSQL 9.5 (localhost:5433) - s2b130050131060 - public.bank

