

## 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.**

### Description:

What the java.sql Package Contains

The java.sql package contains API for the following:

- Making a connection with a database via the DriverManager facility
  - DriverManager class -- makes a connection with a driver
  - SQLPermission class -- provides permission when code running within a Security Manager, such as an applet, attempts to set up a logging stream through the DriverManager
  - Driver interface -- provides the API for registering and connecting drivers based on JDBC technology ("JDBC drivers"); generally used only by the DriverManager class
  - DriverPropertyInfo class -- provides properties for a JDBC driver; not used by the general user
- Sending SQL statements to a database
  - Statement -- used to send basic SQL statements
  - PreparedStatement -- used to send prepared statements or basic SQL statements (derived from Statement)
  - CallableStatement -- used to call database stored procedures (derived from PreparedStatement)
  - Connection interface -- provides methods for creating statements and managing connections and their properties
  - Savepoint -- provides savepoints in a transaction
- Retrieving and updating the results of a query
  - ResultSet interface
- Standard mappings for SQL types to classes and interfaces in the Java programming language
  - Array interface -- mapping for SQL ARRAY
  - Blob interface -- mapping for SQL BLOB
  - Clob interface -- mapping for SQL CLOB
  - Date class -- mapping for SQL DATE
  - NClob interface -- mapping for SQL NCLOB
  - Ref interface -- mapping for SQL REF

- RowId interface -- mapping for SQL ROWID
- Struct interface -- mapping for SQL STRUCT
- SQLXML interface -- mapping for SQL XML
- Time class -- mapping for SQL TIME
- Timestamp class -- mapping for SQL TIMESTAMP
- Types class -- provides constants for SQL types
- Custom mapping an SQL user-defined type (UDT) to a class in the Java programming language
  - SQLData interface -- specifies the mapping of a UDT to an instance of this class
  - SQLInput interface -- provides methods for reading UDT attributes from a stream
  - SQLOutput interface -- provides methods for writing UDT attributes back to a stream
- Metadata
  - DatabaseMetaData interface -- provides information about the database
  - ResultSetMetaData interface -- provides information about the columns of a ResultSet object
  - ParameterMetaData interface -- provides information about the parameters to PreparedStatement commands
- Exceptions
  - SQLException -- thrown by most methods when there is a problem accessing data and by some methods for other reasons
  - SQLWarning -- thrown to indicate a warning
  - DataTruncation -- thrown to indicate that data may have been truncated
  - BatchUpdateException -- thrown to indicate that not all commands in a batch update executed successfully

## Program code:

```
import java.sql.*;
import java.util.*;

class JC6{
    public static void main(String[] args){
        Connection c=null;
        PreparedStatement p=null;
        try{
            Class.forName("org.postgresql.Driver");

            c=DriverManager.getConnection("jdbc:postgresql://localhost:5432/s2b130050131031","postgres","1014");
            Scanner sc =new Scanner(System.in);
            int i=0;
```

```

        System.out.println("130050131031");
        while(i!=1){
            System.out.println("select the option:");
            System.out.println("1.Insert \t 2.Update \n 3.Select \t
4.Delete \t 5.exit");
            int op=sc.nextInt();
            int id,balance;
            String number=null;
            String name=null;
            String add=null;

            switch(op){
                case 1:
                    p=c.prepareStatement("insert into bank
values(?,?,?,?,?)");
                    System.out.print("enter AccountNo:");
                    id=sc.nextInt();
                    System.out.print("enter Customername:");
                    name=sc.next();
                    System.out.print("enter Balance:");
                    balance=sc.nextInt();
                    System.out.print("enter PhoneNo:");
                    number=sc.next();
                    System.out.print("enter Address:");
                    add=sc.next();
                    p.setInt(1,id);
                    p.setString(2,name);
                    p.setInt(3,balance);
                    p.setString(4,number);
                    p.setString(5,add);
                    p.executeUpdate();

                    break;
                case 2:
                    p=c.prepareStatement("update bank set balance=?
where accountno=?");
                    System.out.print("enter accountno:");
                    id=sc.nextInt();
                    System.out.print("enter Balance:");
                    balance=sc.nextInt();
                    p.setInt(3,id);
                    p.setInt(1,balance);
                    p.executeUpdate();
                    break;
                case 3:
                    p=c.prepareStatement("select * from bank where
balance>?");
                    System.out.print("balance greater then:");
                    balance=sc.nextInt();
                    p.setInt(1,balance);
                    ResultSet r=p.executeQuery();
                    while(r.next()){
                        System.out.println(r.getString("accountno")
+"\t"+r.getString("customername")+"\t"+r.getString("balance")+"\t"+
r.getString("phoneno")
+"\t"+r.getString("address"));
                    }
                    break;
                case 4:
                    p=c.prepareStatement("delete from bank where
accountno=?");

```

```

        System.out.print("enter accountno:");
        id=sc.nextInt();
        p.setInt(1,id);
        p.executeUpdate();
        break;
    case 5:
        i=1;
        break;
    default:
        System.out.println("enter correct choice");
        break;
    }
}

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

```

### Input Output:

```
C:\Windows\System32\cmd.exe
E:\java\jdbc>java JC6
130050131031
select the option:
1.Insert          2.Update
3.Select          4.Delete          5.exit
1
enter AccountNo:13557
enter Customername:abhi
enter Balance:14000
enter PhoneNo:9898123312
enter Address:surat
select the option:
1.Insert          2.Update
3.Select          4.Delete          5.exit
3
balance greater then:13000
13557  abhi  14000  9898123312      surat
select the option:
1.Insert          2.Update
3.Select          4.Delete          5.exit
5
E:\java\jdbc>
```