# Practical No:4

# Remote Object Communication

**a) Using MySQL create Library database. Create table Book (Book_id, Book_name, Book_author) and retrieve the Book information from Library database using Remote Object Communication concept.**

**Code:-**

**LibraryDbInf.java:-**

package libraryManagement;

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface LibraryDbInf extends Remote {

        public String getData(String StrQuery) throws RemoteException;

}


**LibraryOperations.java:**

package libraryManagement;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

public class LibraryOperations extends UnicastRemoteObject implements LibraryDbInf {

        Connection con;

        Statement stmt;

        ResultSet rs;

        ResultSetMetaData rsmd;

        String colStr,resultStr;

        private static final long serialVersionUID = 1L;

        protected LibraryOperations() throws RemoteException {

                super();

                colStr="";

```
        resultStr="";

        con=null;

        stmt=null;

        rs=null;

        rsmd=null;

}
//set Database connection

        public void setDbConn()

        {

                try

                {

                        //protocol  //server //portNo //Database

                        String url="jdbc:mysql://localhost:3306/library";

                        //loading of driver

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

                        //get connection

                         con=DriverManager.getConnection(url,"root","");

                }catch(Exception e) {

                        System.out.println(e.getMessage());

                }

        }
@Override
public String getData(String StrQuery) {

        try

        {

                setDbConn();

                System.out.println("Database Connected");

                //create Statement

                stmt=con.createStatement();

                rs=stmt.executeQuery(StrQuery);

                //getting metadata info about result set;

                rsmd=rs.getMetaData();
```

# Practical No:4

```
                //getting the columns of table which is stored in rsmd

                for(int i=1;i<=rsmd.getColumnCount();i++)

                {

                        System.out.println(i);

                        colStr=colStr + rsmd.getColumnName(i)+"\t\t\t"

                }

                //getting all records from result set

                while(rs.next())

                {

                        for(int i=1;i<=rsmd.getColumnCount();i++)

                        {

                                resultStr=resultStr + rs.getString(i)+ "\t\t";

                        }

                        resultStr=resultStr + "\n";

                }}

        catch(Exception e) {

                System.out.println(e.getMessage());

        }

        return colStr + "\n" +resultStr;

}}
```

**Library_Server.java:-**

```
package libraryManagement;

import java.rmi.Naming;

import java.rmi.registry.LocateRegistry;

public class Library_Server {

        public static void main(String[] args) {

                try

                {

                        LibraryDbInf skeleton=new LibraryOperations();

                        //create registry

                        LocateRegistry.createRegistry(1902);
```

# Practical No:4

```
            Naming.rebind("rmi://localhost:1902/libraryServices", skeleton);

            System.out.println("Server Started......");

    }

    catch(Exception e)

    {

            System.out.println("error from 1st catch"+e.getMessage());

    }}}
```

**Library_Client.java:-**

```
package libraryManagement;

import java.rmi.Naming;

public class Library_Client {

    public static void main(String[] args) {

        String sql="";

        try

        {

            LibraryDbInf stub=(LibraryDbInf)
Naming.lookup("rmi://localhost:1902/libraryServices");

            sql="select * from book";

            sql=stub.getData(sql);

            System.out.println(sql);

        }

        catch (Exception e) {

            System.out.println(e.getMessage());

        }}}
```
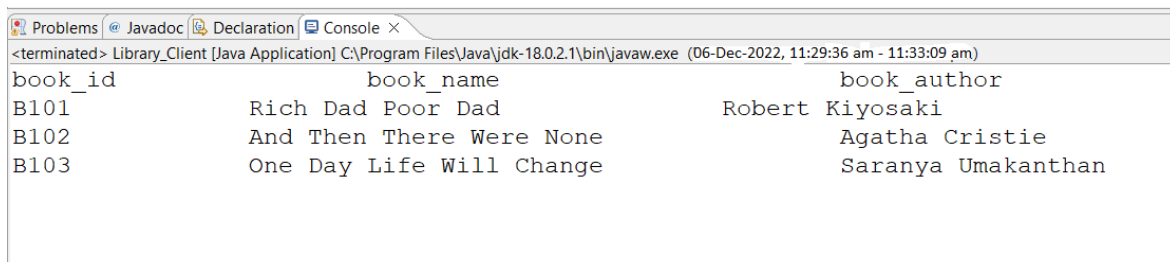
**Output:-**

```
Problems  @ Javadoc  Declaration  Console ×
<terminated> Library_Client [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe  (06-Dec-2022, 11:29:36 am - 11:33:09 am)
book_id              book_name                    book_author
B101        Rich Dad Poor Dad          Robert Kiyosaki
B102        And Then There Were None        Agatha Cristie
B103        One Day Life Will Change        Saranya Umakanthan
```

**b) Using MySQL create Electric_Bill database. Create table Bill (consumer_name, bill_due_date, bill_amount) and retrieve the bill information from the Electric_Bill database using Remote Object Communication concept.**

# Practical No:4

**Code:-**

**ElectricityDbInf.java:-**

package Electricity_Bill_Management;

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface ElectricityDbInf extends Remote {

      public String getData(String StrQuery) throws RemoteException;

}

**ElectricityDbOperations.java:-**

package Electricity_Bill_Management;


import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;


public class ElectrictyDbOperations extends UnicastRemoteObject implements ElectricityDbInf {

      Connection con;

      Statement stmt;

      ResultSet rs;

      ResultSetMetaData rsmd;

      String colStr,resultStr;

      private static final long serialVersionUID = 1L;


      protected ElectrictyDbOperations() throws RemoteException {

            super();

            colStr="";

```java
        resultStr="";

        con=null;

        stmt=null;

        rs=null;

        rsmd=null;

}

public void setDbConn()

{

        try

        {

                //protocol  //server //portNo //Database

                String url="jdbc:mysql://localhost:3306/electric_bill";

                //loading of driver

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

                //get connection

                 con=DriverManager.getConnection(url,"root","");



        }catch(Exception e) {

                System.out.println(e.getMessage());

        }



}

@Override

public String getData(String StrQuery) throws RemoteException {

        try

        {

                setDbConn();

                System.out.println("Database Connected");

                //create Statement

                stmt=con.createStatement();

                rs=stmt.executeQuery(StrQuery);
```

# Practical No:4

```java
            //getting metadata info about result set;

            rsmd=rs.getMetaData();

            //getting the columns of table which is stored in rsmd

            for(int i=1;i<=rsmd.getColumnCount();i++)

            {

                    System.out.println(i);

                    colStr=colStr + rsmd.getColumnName(i)+"\t";

            }

            //getting all records from result set

            while(rs.next())

            {

                    for(int i=1;i<=rsmd.getColumnCount();i++)

                    {

                            resultStr=resultStr + rs.getString(i)+ "\t\t";

                    }

                    resultStr=resultStr + "\n";


            }

        }

        catch(Exception e) {

                System.out.println(e.getMessage());

        }

        return colStr + "\n" +resultStr;

    }}
```

**Electricity_Server.java:-**

```java
package Electricity_Bill_Management;

import java.rmi.Naming;

import java.rmi.registry.LocateRegistry;
```

# Practical No:4

```java
public class Electricity_Server {

    public static void main(String[] args) {

        try

        {

            ElectricityDbInf skeleton=new ElectrictyDbOperations()

            //create registry

            LocateRegistry.createRegistry(1904);

            Naming.rebind("rmi://localhost:1904/electricityServices", skeleton);

            System.out.println("Server Started......");

        }

        catch(Exception e)

        {

            System.out.println("error from 1st catch"+e.getMessage());

        }}}
```

**Electricity_Client.java:-**

```java
package Electricity_Bill_Management;

import java.rmi.Naming;

public class Electricity_Client {

    public static void main(String[] args) {

        String sql="";

        try

        {

            ElectricityDbInf stub=(ElectricityDbInf)
Naming.lookup("rmi://localhost:1904/electricityServices");

            sql="select * from bill";

            sql=stub.getData(sql);

            System.out.println(sql);

        }

        catch (Exception e) {

            System.out.println(e.getMessage());


        }}}
```

# Practical No:4

**Output:-**

```
consumer_name      bill_due_date      bill_amount
akshay             2022-10-26                    1340.65
Rutuja             2022-10-22             •       789.65
Atharva            2022-10-27                     550.0
Simran             2022-10-24                     678.76
```