

## **Distributed Systems and Cloud Computing**

### **Module 4 - Remote Object Communication**

**Program 1: Using MySQL create Library database. Create table Book (Book\_id, Book\_name, Book\_author)**

**Theory:** A library database is essential for managing and organizing book-related information in a systematic manner. Using MySQL, a widely-used relational database management system, we can create a structured database to store data about books. The Book table will include essential fields such as Book\_id, which serves as a unique identifier for each book; Book\_name, which stores the title of the book; and Book\_author, which contains the name of the author. This structured approach enables efficient data retrieval, updates, and management of library resources, facilitating better service delivery and record-keeping in a library environment. By implementing a database, libraries can streamline operations such as tracking book availability and managing user interactions.

**Code:**

**Book table:**

```
create database Library;
use Library;
create table Book(Book_id int, Book_name Varchar(30),Book_author Varchar(30));
insert into Book values(1,"xyz","ABC");
insert into Book values(2,"XYZ","abc");
insert into book values(1,'Harry Potter', 'J.K Rowling');
```

**MyInterface.java**

```
package pack;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface MyInterface extends Remote {
public String getData() throws RemoteException;
}
```

### MyClient.java

```
package pack;
import java.rmi.Naming;
public class MyClient {
public static void main(String args[]) throws Exception
{
    MyInterface obj=(MyInterface)Naming.lookup("rmi://localhost:2099/library");
    String s=obj.getData();
    System.out.println("\n");
    System.out.println(s);

}
}
```

### Register.java

```
package pack;
import java.rmi.Naming;
import java.rmi.Remote;
import java.rmi.registry.LocateRegistry;
public class Register {
public static void main(String[] args) {
// TODO Auto-generated method stub
try

{
    Remote reg=LocateRegistry.createRegistry(2099);
    MyServer obj=new MyServer();
    Naming.rebind("rmi://localhost:2099/library",obj);
}
catch(Exception e)
{
    System.out.println(e);
}
}
}
```

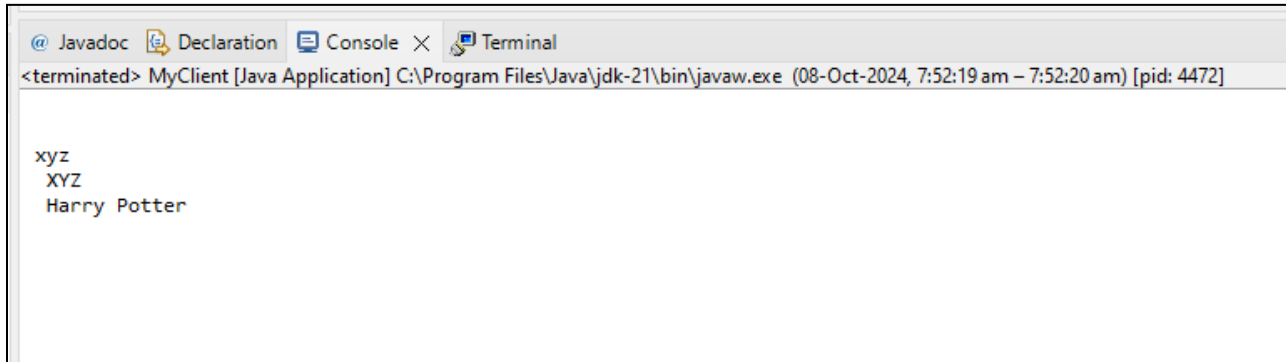
**MyServer.java:**

```
package pack;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class MyServer extends UnicastRemoteObject implements MyInterface
{
    String str="";
    MyServer() throws RemoteException
    {
        super();
    }
    public String getData()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library","root","root");

            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from Book");
            while(rs.next())
            {
                str+=" "+rs.getString(2)+"\n ";
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return str;
    }
}
```

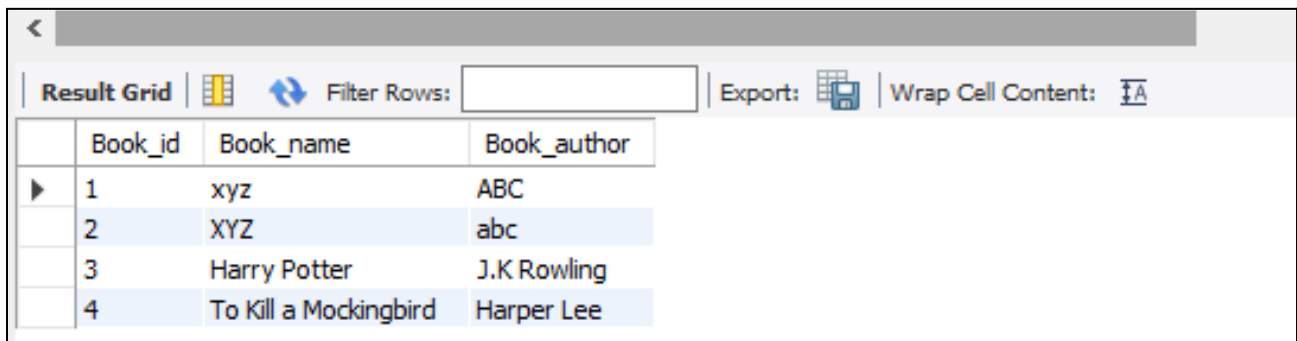
### Output:

We first execute the register and then we execute the MyClient



The screenshot shows a Java IDE with a terminal window titled "MyClient [Java Application]". The terminal output displays the following text:

```
xyz  
XYZ  
Harry Potter
```



The screenshot shows a "Result Grid" table with the following data:

	Book_id	Book_name	Book_author
▶	1	xyz	ABC
	2	XYZ	abc
	3	Harry Potter	J.K Rowling
	4	To Kill a Mockingbird	Harper Lee

#### **DS & CC Lab Unit 4: Remote Object Communication**

Program 10: Using MySQL create Electric\_Bill database. Create table Bill (consumer\_name, bill\_due\_date, bill\_amount)

- 1.retrieve the Bill information from the Electric\_Bill database using Remote Object Communication concept.
- 2.Insert the Bill information from the Electric\_Bill database using Remote Object Communication concept.
- 3.Update the Bill information from the Electric\_Bill database using Remote Object Communication concept.
- 4.Delete the Book information from the Library database using Remote Object Communication.

#### **Theory:**

#### **Code:**

#### **Bill Table:**

```
CREATE DATABASE Electric_Bill;  
USE Electric_Bill;
```

```
CREATE TABLE Bill (  
    consumer_name VARCHAR(255),  
    bill_due_date DATE,  
    bill_amount DECIMAL(10, 2)  
);
```

```
INSERT INTO Bill (consumer_name, bill_due_date, bill_amount) VALUES ('Siddhi Kotre',  
'2024-10-31', 100.50);
```

```
INSERT INTO Bill (consumer_name, bill_due_date, bill_amount) VALUES ('Merideth Grey',  
'2024-11-15', 75.75);
```

```
INSERT INTO Bill (consumer_name, bill_due_date, bill_amount) VALUES ('Mark Sloan',  
'2024-11-01', 200.00);
```

```
select * from Bill;
```

### MyInterface.java

```
package remote;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface MyInterface extends Remote {
    String getBillData() throws RemoteException; // Retrieve bill information
    void insertBill(String consumerName, String dueDate, double amount) throws RemoteException; // Insert new bill
    void updateBill(String consumerName, String dueDate, double amount) throws RemoteException; // Update existing bill
    void deleteBill(String consumerName) throws RemoteException; // Delete bill
}
```

### MyServer.java

```
package remote;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class MyServer extends UnicastRemoteObject implements MyInterface {
    private static final long serialVersionUID = 1L;
    MyServer() throws RemoteException {
        super();
    }
    @Override
    public String getBillData() throws RemoteException {
        StringBuilder str = new StringBuilder();
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Electric_Bill", "root", "root");
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("SELECT * FROM Bill");
            while (rs.next()) {
                str.append(rs.getString("consumer_name")).append(", ")
                    .append(rs.getDate("bill_due_date")).append(", ")
                    .append(rs.getDouble("bill_amount")).append("\n");
            }
            rs.close();
            st.close();
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
        return str.toString();
    }
}
```

```
@Override
public void insertBill(String consumerName, String dueDate, double amount) throws RemoteException {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Electric_Bill", "root",
"root");
        Statement st = con.createStatement();
        String query = "INSERT INTO Bill (consumer_name, bill_due_date, bill_amount) VALUES ('"
            + consumerName + "', '" + dueDate + "', " + amount + ")";
        st.executeUpdate(query);
        st.close();
        con.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}

@Override
public void updateBill(String consumerName, String dueDate, double amount) throws RemoteException {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Electric_Bill", "root",
"root");
        Statement st = con.createStatement();
        String query = "UPDATE Bill SET bill_due_date='" + dueDate + "', bill_amount=" + amount
            + " WHERE consumer_name='" + consumerName + "'";
        st.executeUpdate(query);
        st.close();
        con.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}

@Override
public void deleteBill(String consumerName) throws RemoteException {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Electric_Bill", "root",
"root");
        Statement st = con.createStatement();
        String query = "DELETE FROM Bill WHERE consumer_name='" + consumerName + "'";
        st.executeUpdate(query);
        st.close();
        con.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}
}
```

### MyClient.java

```
package remote;
import java.rmi.Naming;
public class MyClient {
    public static void main(String args[]) throws Exception {
        MyInterface obj = (MyInterface) Naming.lookup("rmi://localhost:2099/Electric_Bill");
        // Retrieve bill data
        String bills = obj.getBillData();
        System.out.println("Bill Information:\n" + bills);
        // Insert new bill
        obj.insertBill("Bob Brown", "2024-12-01", 150.00);
        System.out.println("Inserted new bill for Bob Brown.");
        // Update existing bill
        obj.updateBill("John Doe", "2024-10-30", 120.00);
        System.out.println("Updated bill for John Doe.");
        // Delete a bill
        obj.deleteBill("Jane Smith");
        System.out.println("Deleted bill for Jane Smith.");
    }
}
```

### Register.java

```
package remote;
import java.rmi.Naming;
import java.rmi.Remote;
import java.rmi.registry.LocateRegistry;
public class Register {
    public static void main(String[] args) {
        try {
            Remote reg = LocateRegistry.createRegistry(2099);
            MyServer obj = new MyServer();
            Naming.rebind("rmi://localhost:2099/Electric_Bill", obj);
            System.out.println("Server is ready.");
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



**Output:**

Result Grid			
	consumer_name	bill_due_date	bill_amount
▶	Siddhi Kotre	2024-10-31	100.50
	Merideth Grey	2024-11-15	75.75
	Mark Sloan	2024-11-01	200.00

Problems @ Javadoc Declaration Console × Progress Terminal TestNG  
Register [Java Application] C:\Users\Siddhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotsp  
Server is ready.

Problems @ Javadoc Declaration Console × Progress Terminal TestNG  
<terminated> MyClient [Java Application] C:\Users\Siddhi\.p2\pool\plugins\org.eclipse.justj.o  
Bill Information:  
  
Inserted new bill for Bob Brown.  
Updated bill for Siddhi Kotre.  
Deleted bill for Merideth Grey.