

Assignment No	9
Title	Product Application
Objective	To fetch & display details from the Database
Roll No	MCA2511

**Product.java**

```
package edu.met.p1;
```

```
public class Product {
    int pid;
    String pname;
    String pdesc;
    int price;
    public Product() {
        super();
    }
    public Product(int pid, String pname, String pdesc, int price) {
        super();
        this.pid = pid;
        this.pname = pname;
        this.pdesc = pdesc;
        this.price = price;
    }
    public int getPid() {
        return pid;
    }
    public void setPid(int pid) {
        this.pid = pid;
    }
    public String getPname() {
        return pname;
    }
    public void setPname(String pname) {
        this.pname = pname;
    }
}
```

```
}  
public String getPdesc() {  
    return pdesc;  
}  
public void setPdesc(String pdesc) {  
    this.pdesc = pdesc;  
}  
public int getPrice() {  
    return price;  
}  
public void setPrice(int price) {  
    this.price = price;  
}  
@Override  
public String toString() {  
    return "Product [pid=" + pid + ", pname=" + pname + ", pdesc=" + pdesc + ",  
price=" + price + " ]";  
}  
  
}
```

### **ProductDao.java**

```
package edu.met.p1;  
  
import org.springframework.jdbc.core.*;  
import java.util.*;  
  
public class ProductDao {  
    JdbcTemplate jdbcT;  
  
    public ProductDao(JdbcTemplate jdbcT) {  
        super();  
        this.jdbcT = jdbcT;  
    }  
  
    //create a method to fetch all records from products table  
    List<Product> getAllProducts()  
    {  
        String selQ = "select * from myproducts";  
        return (List<Product>) jdbcT.query(selQ, new ProductRowMapper());  
    }  
}
```

}

### **ProductRowMapper.java**

```
package edu.met.p1;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

public class ProductRowMapper implements RowMapper<Product> {

    @Override
    public Product mapRow(ResultSet rs, int arg1) throws SQLException {
        // TODO Auto-generated method stub
        Product p1 = new Product();
        p1.setPid(rs.getInt(1));
        p1.setPname(rs.getString(2));
        p1.setPdesc(rs.getString(3));
        p1.setPrice(rs.getInt(4));

        return p1;
    }
}
```

### **ProductImpl.java**

```
package edu.met.p1;

import org.springframework.context.*;
import org.springframework.context.support.*;
import java.util.*;

public class ProductImpl {

    private static ApplicationContext ctx;
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ctx = new ClassPathXmlApplicationContext("AppCtx.xml");

        //get the dao bean
        ProductDao p1 = (ProductDao)ctx.getBean("productDao");
    }
}
```

```
List<Product> l1 = p1.getAllProducts();

//Traverse through this list
System.out.println("pid \tpname\tdesc \t\tprice ");
System.out.println("-----");
for(Product p:l1)
{
    System.out.print(p.getPid()+"\t");
    System.out.print(p.getPname()+"\t");
    System.out.print(p.getPdesc()+"\t");
    System.out.print(p.getPrice()+"\t\n");
}
System.out.println("-----");

}

}
```

#### **AppCtx.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

    <bean id="ds"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName" value="org.postgresql.Driver"/>
        <property name="url" value="jdbc:postgresql://localhost:5432/postgres"/>
        <property name="username" value="postgres"/>
        <property name="password" value="password"/>
    </bean>

    <bean id="jdbcT" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="ds"/>
    </bean>

    <bean id="product" class="edu.met.p1.Product">
        <constructor-arg value="106"/>
        <constructor-arg value="Watch"/>
        <constructor-arg value="Its Watch"/>
    </bean>
</beans>
```

```

    <constructor-arg value="1000"/>
  </bean>

  <bean id="productDao" class="edu.met.p1.ProductDao">
    <constructor-arg ref="jdbcT"/>
  </bean>
</beans>

```

### Output:

pid	pname	desc	price
101	Shirt	This is shirt	700
102	T-Shirt	This is T-Shirt	400
103	Jeans	This is Jeans	1200
104	Hoddie	This is Hoddie	800
105	Belt	This is Belt	200

	pid	pname	description	price
	integer	character varying (30)	character varying (100)	integer
1	101	Shirt	This is shirt	700
2	102	T-Shirt	This is T-Shirt	400
3	103	Jeans	This is Jeans	1200
4	104	Hoddie	This is Hoddie	800
5	105	Belt	This is Belt	200