

Assignment No	7
Title	Movie Application
Objective	To Insert and Update a row using JDBC
Roll No	MCA2511

1. To Insert and Update a row using JDBC.

Source Code:

MovieBean.java

```
package edu.met.p1;

public class MovieBean {
    int mid;
    String mtitle;
    String Director;
    public int getMid() {
        return mid;
    }
    public void setMid(int mid) {
        this.mid = mid;
    }
    public String getMtitle() {
        return mtitle;
    }
    public void setMtitle(String mtitle) {
        this.mtitle = mtitle;
    }
    public String getDirector() {
        return Director;
    }
    public void setDirector(String director) {
        Director = director;
    }
}
```

```
@Override  
public String toString() {  
    return "MovieBean [mid=" + mid + ", mtitle=" + mtitle + ", Director=" +  
Director + "]";  
}  
}
```

MoviesDAO.java

```
package edu.met.p1;  
  
import org.springframework.jdbc.core.JdbcTemplate;  
  
public class MoviesDAO {  
  
    JdbcTemplate jdbcT;  
  
    public MoviesDAO (JdbcTemplate jdbcT) {  
        super();  
        this.jdbcT=jdbcT;  
    }  
    //method to insert a record in table  
    void insertRow(MovieBean m1)  
    {  
        String insertQ = "insert into mymovies values("+m1.getMid()+",""+  
m1.getMtitle()+","+m1.getDirector()+"");  
        jdbcT.update(insertQ);  
    }  
  
    //method to update a record in table  
    void updateRow(MovieBean m1)  
    {  
        String updateQ = "update mymovies set mname = ?, director = ? where  
mid=?";  
        jdbcT.update(updateQ,m1.getMtitle(),m1.getDirector(),m1.getMid());  
    }  
}
```

MoviesImpl.java

```
package edu.met.p1;  
  
import org.springframework.context.ApplicationContext;
```

```
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MoviesImpl {

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

        ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
        //get movies bean Object
        MovieBean m1 = (MovieBean)ctx.getBean("movie");
        // Get the DAO bean object
        MoviesDAO dao = (MoviesDAO)ctx.getBean("moviesDao");
        // insert the record
        //dao.insertRow(m1);
        m1.setMid(101);
        m1.setMtitle("Badlapur Stories");
        m1.setDirector("Omkar");
        //dao.insertRow(m1);
        dao.updateRow(m1);
    }
}
```

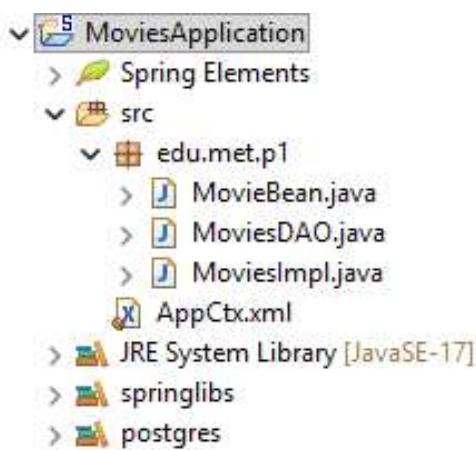
AppCtx.xml

```
<?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="moviesDao" class="edu.met.p1.MoviesDAO">
    <constructor-arg name="jdbcT" ref="jdbcT"/>
</bean>
<bean id="movie" class="edu.met.p1.MovieBean">
    <property name="mid" value="101"/>
    <property name="mtitle" value="KGF"/>
    <property name="director" value="Omkar"/>
</bean>
</beans>
```

Output:



	mid integer	mname character varying (20)	director character varying (20)
1	102	Kantara	Bahubali
2	103	RRR	Vijay
3	104	Muna Bhai	Sanjay
4	105	3 Ediots	Amir
5	201	Western Stories	Aman
6	101	Badlapur Stories	Omkar
7	101	Badlapur Stories	Omkar
8	101	Badlapur Stories	Omkar