Advanced Java

Assignment - 5

Name – Umme Kulsum | Div – A | Roll no -25

1. **Write a program to insert, update and delete records from the given table.**

Movie1Kulsum.java

```
package org.me;
public class Movie1Kulsum{
int mid;String title;String actor;
public Movie1Kulsum(int mid,String title,String actor){
super();
this.mid=mid;
this.title=title;
this.actor=actor;
}
public Movie1Kulsum(){super();}
public int getMid(){return mid;}
public void setMid(int mid){this.mid=mid;}
public String getTitle(){return title;}
public void setTitle(String title){this.title=title;}
public String getActor(){return actor;}
public void setActor(String actor){this.actor=actor;}
}
```

MovieDAOKulsum.java

```
package org.me;
import org.springframework.jdbc.core.JdbcTemplate;
public class MovieDAOKulsum{
JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){
this.jdbcTemplate=jdbcTemplate;
}
public int insMovie(Movie1Kulsum m1){
String insSql="insert into mymovies1
values("+m1.getMid()+",'"+m1.getTitle()+"','"+m1.getActor()+"')";
return jdbcTemplate.update(insSql);
}
public int updateMovie(Movie1Kulsum m1){
String query="update mymovies1 set title='"+m1.getTitle()+"',actor='"+m1.getActor()+"' where
mid='"+m1.getMid()+"'";
return jdbcTemplate.update(query);
}
public int deleteMovie(Movie1Kulsum m1){
String query="delete from mymovies1 where mid='"+m1.getMid()+"'";
return jdbcTemplate.update(query);
}
}
```

appctx.xml

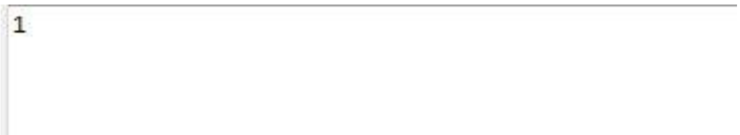
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<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="admin"/>
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"/>
```

```
</bean>
<bean id="mymovie" class="org.me.MovieDAOKulsum">
<property name="jdbcTemplate" ref="jdbcTemplate"/> </bean> </beans>
```

MovieTestKulsum.java

```
package org.me;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTestKulsum{
private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx.xml");
MovieDAOKulsum m1=(MovieDAOKulsum)appCon.getBean("mymovie");
// insert
Movie1Kulsum t1=new Movie1Kulsum(10,"Mirzapur","P");
System.out.println(m1.insMovie(t1));
// update
//System.out.println(m1.updateMovie(new Movie1Kulsum(10,"War","Hrithik")));
// delete
//Movie1Kulsum t2=new Movie1Kulsum();
//t2.setMid(10);
//System.out.println(m1.deleteMovie(t2));
}
}
```

Output



```
1
```

2. Write a program to demonstrate PreparedStatement in Spring JdbcTemplate.

Movie1Kulsum.java

```
package org.me;
public class Movie1Kulsum{
int mid;String title;String actor;
public Movie1Kulsum(int mid,String title,String actor){
super();
this.mid=mid;
this.title=title;
this.actor=actor;
}
public Movie1Kulsum(){super();}
public int getMid(){return mid;}
public void setMid(int mid){this.mid=mid;}
public String getTitle(){return title;}
public void setTitle(String title){this.title=title;}
public String getActor(){return actor;}
public void setActor(String actor){this.actor=actor;}
}
```

MovieDAO1Kulsum.java

```
package org.me;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.PreparedStatementCallback;
```

```
public class MovieDAO1Kulsum{
JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){
this.jdbcTemplate=jdbcTemplate;
}
public Boolean saveMovieByPreparedStatement(final Movie1Kulsum e){
String query="insert into movies values(?,?,?)";
return jdbcTemplate.execute(query,new PreparedStatementCallback<Boolean>(){
@Override
public Boolean doInPreparedStatement(PreparedStatement ps)throws
SQLException,DataAccessException{
ps.setInt(1,e.getMid());
ps.setString(2,e.getTitle());
ps.setString(3,e.getActor());
return ps.execute();
}
});
}
}
```

appctx1.xml

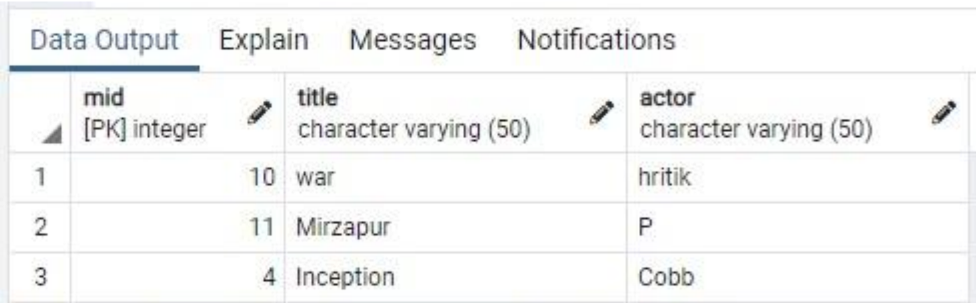
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<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="pass"/>
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
```

```
<property name="dataSource" ref="ds"/>
</bean>
<bean id="mymovie" class="org.me.MovieDAO1Kulsum">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean>
</beans>
```

MovieTest1Kulsum.java

```
package org.me;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest1Kulsum{
private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx1.xml");
MovieDAO1Kulsum m1=(MovieDAO1Kulsum)appCon.getBean("mymovie");
m1.saveMovieByPreparedStatement(new Movie1Kulsum(5,"Bhaijaan","Kulsum"));
}
}
```

Output



	mid [PK] integer	title character varying (50)	actor character varying (50)
1	10	war	hritik
2	11	Mirzapur	P
3	4	Inception	Cobb

3. Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface.

Movie2Kulsum.java

```
package org.me;
public class Movie2Kulsum{
int mid;String title;String actor;
public int getMid(){return mid;}
public void setMid(int mid){this.mid=mid;}
public String getTitle(){return title;}
public void setTitle(String title){this.title=title;}
public String getActor(){return actor;}
public void setActor(String actor){this.actor=actor;}
public String toString(){return mid+" "+title+" "+actor;}
}
```

MovieDAO2Kulsum.java

```
package org.me;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.ResultSetExtractor;
public class MovieDAO2Kulsum{
JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){this.jdbcTemplate=jdbcTemplate;}
public List<Movie2Kulsum> getAllMovie(){
```

```
return jdbcTemplate.query("select * from mymovies1",new
ResultSetExtractor<List<Movie2Kulsum>>(){
@Override
public List<Movie2Kulsum> extractData(ResultSet rs)throws SQLException,DataAccessException{
List<Movie2Kulsum> list=new ArrayList<Movie2Kulsum>();
while(rs.next()){
Movie2Kulsum e=new Movie2Kulsum();
e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3));
list.add(e);
}
return list;
}
});
}
}
```

appctx2.xml

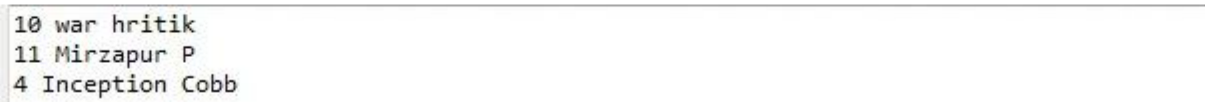
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<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="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"/>
</bean>
```

```
<bean id="mymovie" class="org.me.MovieDAO2Kulsum">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean> </beans>
```

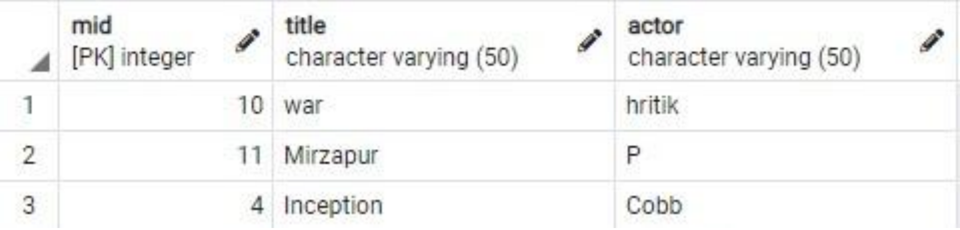
MovieTest2Kulsum.java

```
package org.me;
import java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest2Kulsum{
private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx2.xml");
MovieDAO2Kulsum m1=(MovieDAO2Kulsum)appCon.getBean("mymovie");
List<Movie2Kulsum> list=m1.getAllMovie();
for(Movie2Kulsum e:list)
System.out.println(e);
}
}
```

Output



```
10 war hritik
11 Mirzapur P
4 Inception Cobb
```



	mid [PK] integer	title character varying (50)	actor character varying (50)
1	10	war	hritik
2	11	Mirzapur	P
3	4	Inception	Cobb

4. **Write a program to demonstrate RowMapper interface to fetch the records from the database.**

Movie3Kulsum.java

```
package org.me;

public class Movie3Kulsum{

int mid;String title;String actor;

public Movie3Kulsum(int mid,String title,String actor){

super();

this.mid=mid;

this.title=title;

this.actor=actor;

}

public Movie3Kulsum(){super();}

public int getMid(){return mid;}

public void setMid(int mid){this.mid=mid;}

public String getTitle(){return title;}

public void setTitle(String title){this.title=title;}

public String getActor(){return actor;}

public void setActor(String actor){this.actor=actor;}

public String toString(){return mid+" "+title+" "+actor;}

}
```

MovieDAO3Kulsum.java

```
package org.me;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
public class MovieDAO3Kulsum{
JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){this.jdbcTemplate=jdbcTemplate;}
public List<Movie3Kulsum> getAllEmployeesRowMapper(){
return jdbcTemplate.query("select * from mymovies1",new RowMapper<Movie3Kulsum>(){
@Override
public Movie3Kulsum mapRow(ResultSet rs,int rownumber)throws SQLException{
Movie3Kulsum e=new Movie3Kulsum();
e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3));
return e;
}
});
}
}
```

appctx3.xml

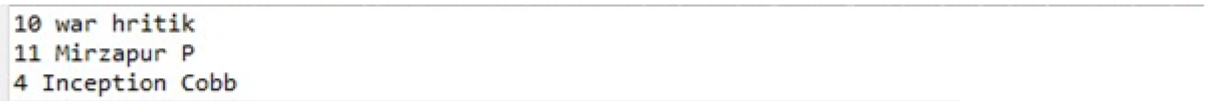
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<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="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"/>
```

```
</bean>
<bean id="mymovie" class="org.me.MovieDAO3Kulsum">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean>
</beans>
```

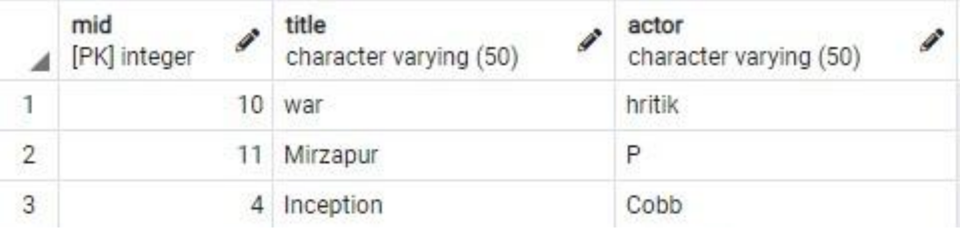
MovieTest3Kulsum.java

```
package org.me;
import java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest3Kulsum{
private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx3.xml");
MovieDAO3Kulsum m1=(MovieDAO3Kulsum)appCon.getBean("mymovie");
List<Movie3Kulsum> list=m1.getAllEmployeesRowMapper();
for(Movie3Kulsum e:list)
System.out.println(e);
}
}
```

Output



```
10 war hritik
11 Mirzapur P
4 Inception Cobb
```



	mid [PK] integer	title character varying (50)	actor character varying (50)
1	10	war	hritik
2	11	Mirzapur	P
3	4	Inception	Cobb