NAME: rohan waghmode CLAS: MCA ROLL NO :67(A)

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

Movie1Atharv.java

```
package org.me; public class
Movie1Atharv{ int mid;String
title;String actor;
public Movie1Atharv(int mid,String title,String actor){
super(); this.mid=mid; this.title=title;
this.actor=actor;
}
public Movie1Atharv(){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;}
}
```

MovieDAOAtharv.java

```
package org.me;
import org.springframework.jdbc.core.JdbcTemplate;
public class MovieDAOAtharv{ JdbcTemplate
jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){ this.jdbcTemplate=jdbcTemplate;
}
public int insMovie(Movie1Atharv m1){
String insSql="insert into mymovies1
values("+m1.getMid()+",'"+m1.getTitle()+"','"+m1.getActor()+"')"; return
jdbcTemplate.update(insSql);
}
public int updateMovie(Movie1Atharv m1){
String query="update mymovies1 set title='"+m1.getTitle()+"',actor='"+m1.getActor()+"' where
mid='"+m1.getMid()+"'"; return jdbcTemplate.update(query);
}
public int deleteMovie(Movie1Atharv 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.MovieDAOAtharv">
<property name="jdbcTemplate" ref="jdbcTemplate"/> </bean> </beans>
```

MovieTestAtharv.java package org.me;

```
import org.springframework.context.ApplicationContext;
import   org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTestAtharv{ private static ApplicationContext appCon; public
static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx.xml");
MovieDAOAtharv m1=(MovieDAOAtharv)appCon.getBean("mymovie");
// insert
Movie1Atharv t1=new Movie1Atharv(10,"Mirzapur","P");
System.out.println(m1.insMovie(t1));
// update
//System.out.println(m1.updateMovie(new  Movie1Atharv(10,"War","Hrithik")));
// delete
//Movie1Atharv t2=new Movie1Atharv();
//t2.setMid(10);
//System.out.println(m1.deleteMovie(t2));
}
}
```

Output

```
1
```

NAME: rohan waghmode CLAS: MCA ROLL NO :67(A)

Write a program to demonstrate PreparedStatement in Spring JdbcTemplate.

Movie1Atharv.java

```
package org.me; public class
Movie1Atharv{ int mid;String
title;String actor;
public Movie1Atharv(int mid,String title,String actor){
super(); this.mid=mid; this.title=title;
this.actor=actor;
}
public Movie1Atharv(){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;}
}
```

MovieDAO1Atharv.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 MovieDAO1Atharv{ JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){ this.jdbcTemplate=jdbcTemplate;
}
public Boolean saveMovieByPreparedStatement(final Movie1Atharv 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();
}
});
}
}
```

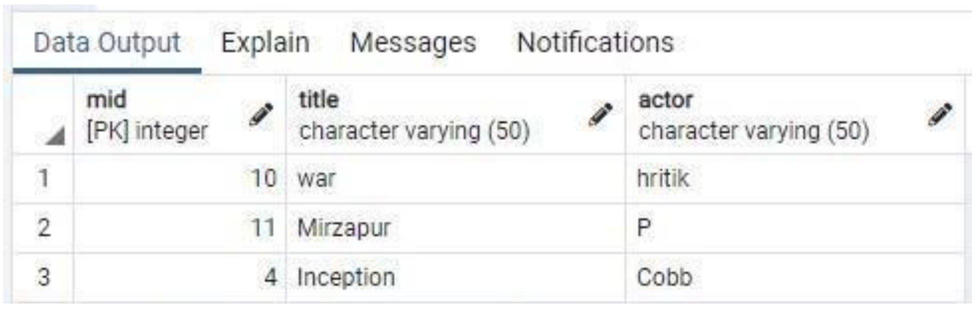
appctx1.xml

```
<?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.MovieDAO1Atharv">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean>
</beans>
```

MovieTest1Atharv.java

```
package org.me;
import org.springframework.context.ApplicationContext;
import    org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest1Atharv{ private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new                              ClassPathXmlApplicationContext("appctx1.xml");
MovieDAO1Atharv            m1=(MovieDAO1Atharv)appCon.getBean("mymovie");
m1.saveMovieByPreparedStatement(new Movie1Atharv(5,"Bhaijaan","Atharv"));
}
}
```

Output



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

NAME: rohan waghmode CLAS: MCA ROLL NO :67(A)

Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface.

Movie2Atharv.java
```
package org.me; public class
Movie2Atharv{ 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;}
}
```

MovieDAO2Atharv.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 MovieDAO2Atharv{ JdbcTemplate
jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){this.jdbcTemplate=jdbcTemplate;}
public List<Movie2Atharv> getAllMovie(){
return jdbcTemplate.query("select * from mymovies1",new
ResultSetExtractor<List<Movie2Atharv>>(){
@Override
public List<Movie2Atharv> extractData(ResultSet rs)throws SQLException,DataAccessException{
List<Movie2Atharv> list=new ArrayList<Movie2Atharv>(); while(rs.next()){
Movie2Atharv e=new Movie2Atharv();
e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3));  list.add(e);
}
return list;
}
});
}
}
```

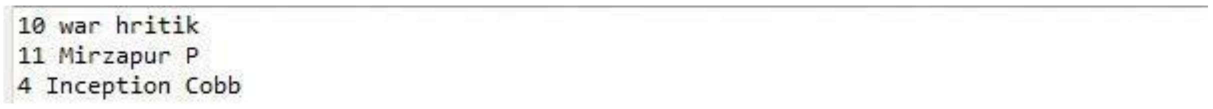
appctx2.xml

```
<?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.MovieDAO2Atharv">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean> </beans>
```

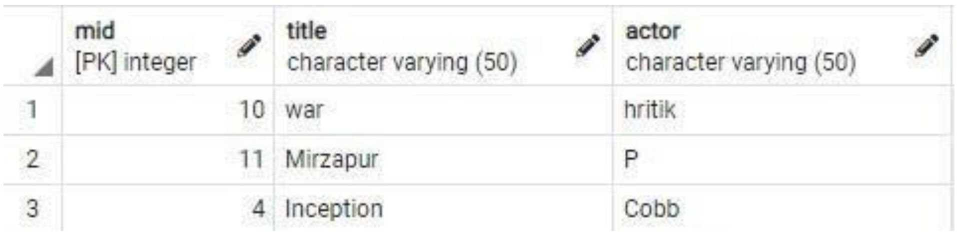
MovieTest2Atharv.java

```
package org.me; import
java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest2Atharv{ private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx2.xml");
MovieDAO2Atharv m1=(MovieDAO2Atharv)appCon.getBean("mymovie");
List<Movie2Atharv> list=m1.getAllMovie(); for(Movie2Atharv 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

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

Movie3Atharv.java

```
package org.me; public class
Movie3Atharv{ int mid;String
title;String actor;
public Movie3Atharv(int mid,String title,String actor){
super(); this.mid=mid;
this.title=title; this.actor=actor;
}
public Movie3Atharv(){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;}
}
```

MovieDAO3Atharv.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 MovieDAO3Atharv{
JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate){this.jdbcTemplate=jdbcTemplate;}
public List<Movie3Atharv> getAllEmployeesRowMapper(){
return jdbcTemplate.query("select * from mymovies1",new RowMapper<Movie3Atharv>(){
@Override
public Movie3Atharv mapRow(ResultSet rs,int rownumber)throws SQLException{
Movie3Atharv e=new Movie3Atharv();
e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3)); return
e;
}
});
}
}
```

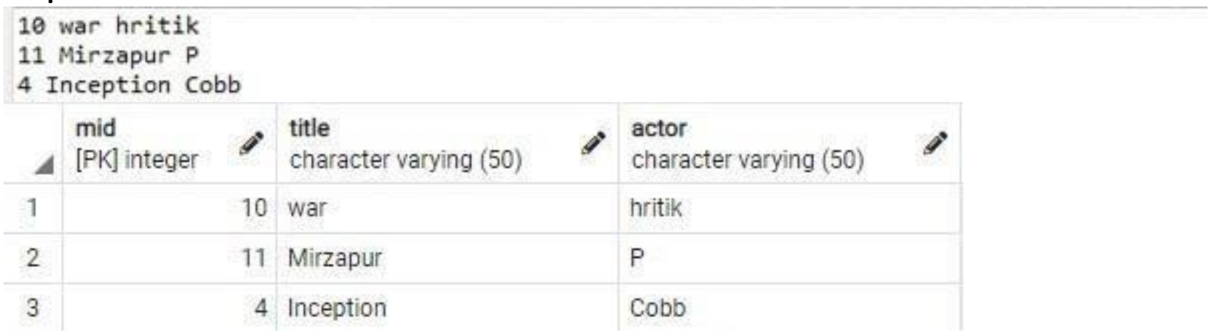
appctx3.xml

```
<?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.MovieDAO3Atharv">
<property name="jdbcTemplate" ref="jdbcTemplate"/>
</bean>
</beans>
```

MovieTest3Atharv.java

```
package org.me; import
java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MovieTest3Atharv{ private static ApplicationContext appCon;
public static void main(String[] args){
appCon=new ClassPathXmlApplicationContext("appctx3.xml");
MovieDAO3Atharv m1=(MovieDAO3Atharv)appCon.getBean("mymovie");
List<Movie3Atharv> list=m1.getAllEmployeesRowMapper(); for(Movie3Atharv
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