**Assignment 4**

1. **Write a program to insert, update and delete records from the Movie table.(Movie Application) Movie table(movie\_id , movie\_name)**

**App.java**

package module4;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext; public class App {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext con = new ClassPathXmlApplicationContext("config.xml"); MovieDAO m\_dao = con.getBean("movieDAO", MovieDAO.class);

// Insert Values in Table

Movie m = new Movie(1, "Fireflies"); Movie m1 = new Movie(2, "Moana"); Movie m2 = new Movie(3, "Frozen");

System.out.println("Data Inserted Successfully : " + m\_dao.insertMovie(m)); System.out.println("Data Inserted Successfully : " + m\_dao.insertMovie(m1)); System.out.println("Data Inserted Successfully : " + m\_dao.insertMovie(m2));

// Update Values in Table

Movie m3 = new Movie(2, "Frozen II");

System.out.println("Table Updated Successfully : " + m\_dao.UpdateMovie(m3));

// DELETE RECORD FROM TABLE

System.out.println("\n Delete record : " + m\_dao.deleteMovie(1));

}

}

**Movie.java**

**package** module4;

**public class** Movie {

**private int** id;

**private** String movie\_name;

**public int** getId() {

**return** id;

}

**public void** setId(**int** id) {

**this**.id = id;

}

**public** String getMovie\_name() {

**return** movie\_name;

}

**public void** setMovie\_name(String movie\_name) {

**this**.movie\_name = movie\_name;

}

**public** Movie(**int** id, String movie\_name) {

**super**(); **this**.id = id;

**this**.movie\_name = movie\_name;

}

**public** Movie() {

**super**();

// **TODO** Auto-generated constructor stub

}

}

**MovieDAO.java**

**package** module4;

**import** org.springframework.jdbc.core.JdbcTemplate;

**public class** MovieDAO {

**private** JdbcTemplate jdbcTemplate;

**public void** setJdbcTemplate(JdbcTemplate jdbcTemplate) {

**this**.jdbcTemplate = jdbcTemplate;

}

**public int** insertMovie(Movie m1) {

// insert into Movie values (1,'Fireflies');

String query = "insert into Movie values (" + m1.getId() + ",'" + m1.getMovie\_name() + "')";

**return** jdbcTemplate.update(query);

}

**public int** deleteMovie(**int** id) {

String query = "DELETE FROM Movie WHERE id =" + id; System.***out***.print(query);

**return** jdbcTemplate.update(query);

}

**public int** UpdateMovie(Movie m3) {

m3.getId() + "'";

}

}

**config.xml**

String query = "update Movie set name='" + m3.getMovie\_name() + "' where id='" +

**return** jdbcTemplate.update(query);

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd) *"*>

<bean id=*"ds"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/film"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"sdf@1073"*></property>

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"ds"* />

</bean>

<bean id=*"movieDAO"* class=*"module4.MovieDAO"*>

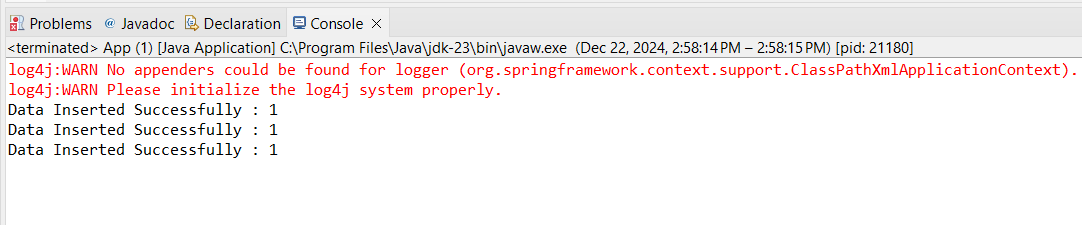
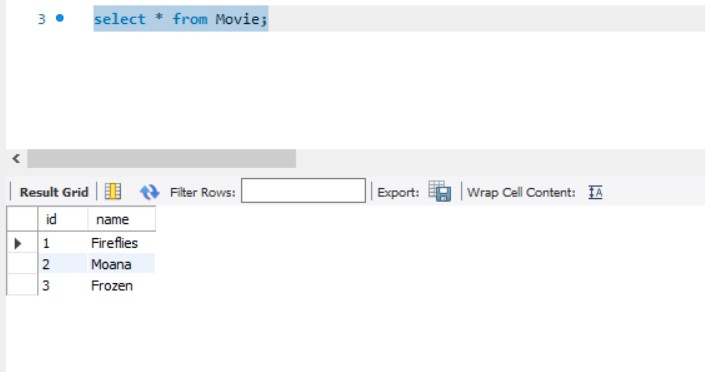
<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*></property>

</bean>

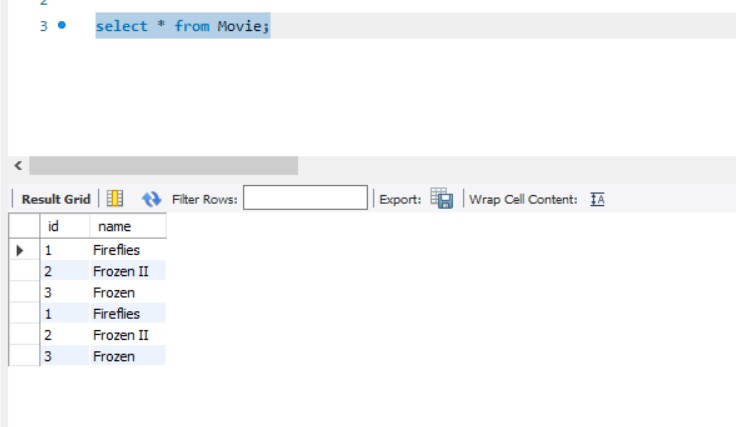
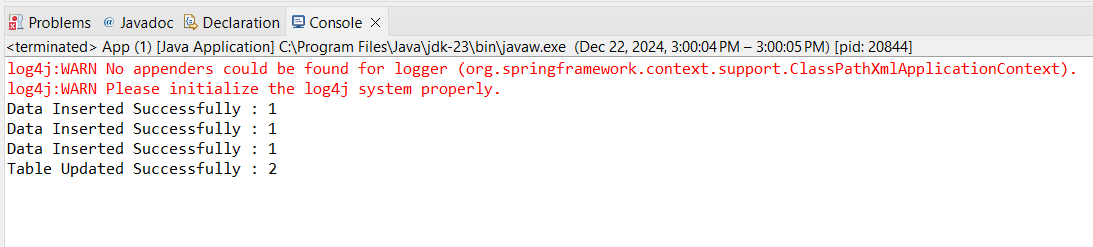
</beans>

**Output:-**

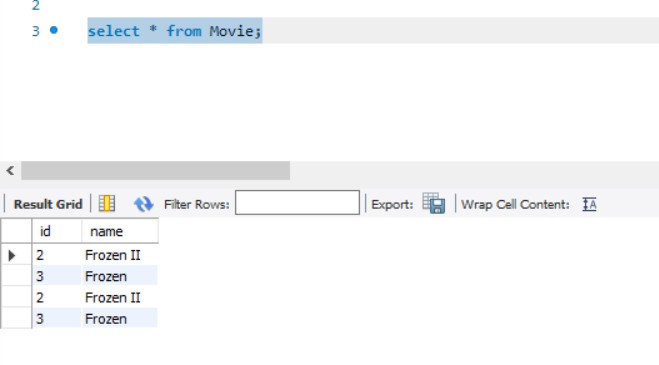
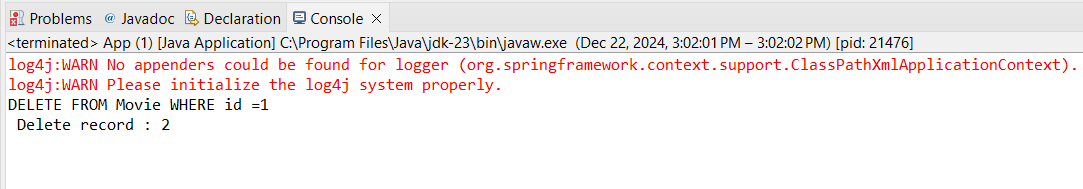
**Insert Record in Movie Table:**

****

**Update Record in Movie Table:-**

****

**Delete Record From Table:**

****

1. **Write a program to demonstrate PreparedStatement in Spring JdbcTemplate.**

**App.java**

package demo.college; import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext; public class App {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext context=new ClassPathXmlApplicationContext("config.xml"); EmployeeDAO edao=context.getBean("e\_dao",EmployeeDAO.class);

Employee e=new Employee(1,"Sandhya", "[Sand@gmail.com](mailto:Sand@gmail.com)"); System.out.println(edao.addEmployee(e));

}

}

**Employee.java package** demo.college; **public class** Employee {

**private int** id; **private** String name; **private** String email;

**public int** getId() {

**return** id;

}

**public void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public void** setEmail(String email) {

**this**.email = email;

}

**public** Employee() {

**super**();

}

**public** Employee(**int** id, String name, String email) {

**this**.id = id; **this**.name = name; **this**.email = email;

}

}

**EmployeeDAO.java**

package demo.college;

import java.sql.PreparedStatement; 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.PreparedStatementCallback; import org.springframework.jdbc.core.ResultSetExtractor;

public class EmployeeDAO {

private JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate = jdbcTemplate;

}

public Boolean addEmployee (final Employee e) {

String query="insert into Employee values(?,?,?)";

return jdbcTemplate.execute(query,new PreparedStatementCallback<Boolean>()

{

@Override

public Boolean doInPreparedStatement(PreparedStatement arg0) throws SQLException, DataAccessException {

arg0.setInt(1, e.getId()); arg0.setString(2, e.getName()); arg0.setString(3, e.getEmail()); return arg0.execute();

}

});

}

}

**Config.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd) *"*>

<bean id=*"ds"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/college"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"sdf @1073"*></property>

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"ds"*></property>

</bean>

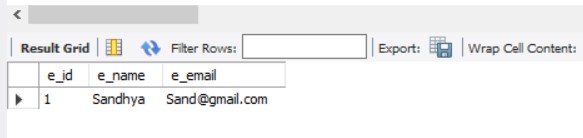
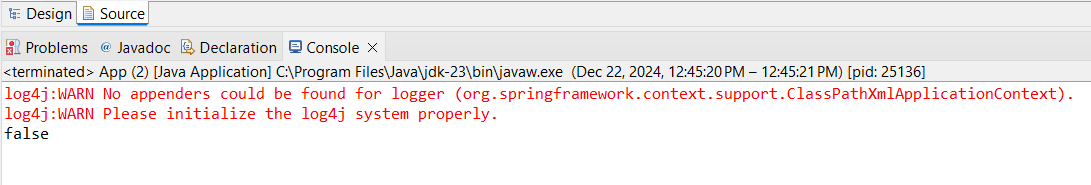
<bean id=*"e\_dao"* class=*"demo.college.EmployeeDAO"*>

<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*></property>

</bean>

</beans>

# Output:-

****

1. **Write a program to demonstrate the ResultSetExtractor interface to fetch the records from the database.**

**App.java**

package demo.college; import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext; public class App {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext context = new ClassPathXmlApplicationContext("config.xml"); EmployeeDAO edao = context.getBean("e\_dao", EmployeeDAO.class);

List<Employee> emp\_list = edao.getAllEmployees(); System.out.println("Data Extracted:");

for (Employee e : emp\_list) { System.out.println("Id:" + e.getId()); System.out.println("Name:" + e.getName()); System.out.println("Email:" + e.getEmail()); System.out.println();

}

}

}

**Employee.java**

**package** demo.college;

**public class** Employee {

**private int** id; **private** String name; **private** String email; **public int** getId() {

**return** id;

}

**public void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public void** setEmail(String email) {

**this**.email = email;

}

**public** Employee() {

**super**();

}

**public** Employee(**int** id, String name, String email) {

**this**.id = id; **this**.name = name; **this**.email = email;

}

}

**EmployeeDAO.java**

**package** demo.college; **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** EmployeeDAO {

**private** JdbcTemplate jdbcTemplate;

**public void** setJdbcTemplate(JdbcTemplate jdbcTemplate) {

**this**.jdbcTemplate = jdbcTemplate;

}

**public** List<Employee> getAllEmployees(){ String query="select \* from Employee";

**return** jdbcTemplate.query(query, **new** ResultSetExtractor<List<Employee>>() {

DataAccessException{

@Override

**public** List<Employee> extractData(ResultSet arg0) **throws** SQLException,

List<Employee> employees=**new** ArrayList<Employee>();

**while**(arg0.next() ) {

Employee e= **new** Employee(); e.setId(arg0.getInt(1)); e.setName(arg0.getString(2)); e.setEmail(arg0.getString(3)); employees.add(e);

}

});

}

}

}

**return** employees;

**Config.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd) *"*>

<bean id=*"ds"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/college"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"sdf@1073"*></property>

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"ds"*></property>

</bean>

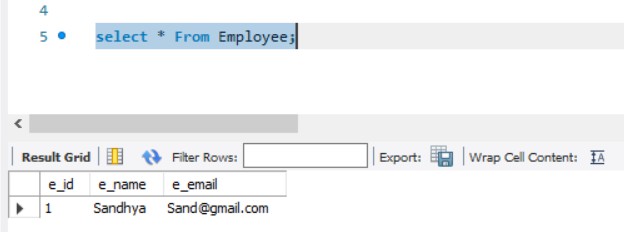
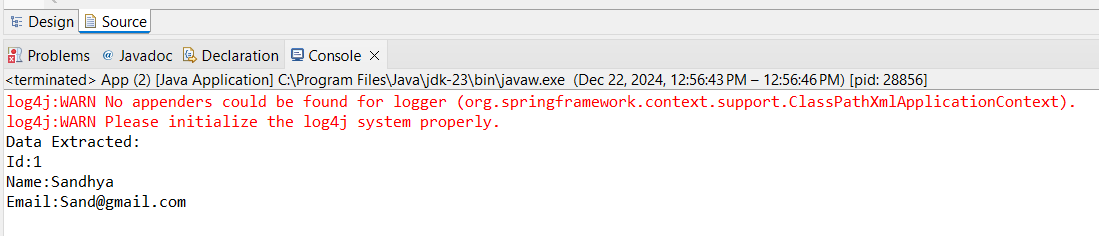
<bean id=*"e\_dao"* class=*"demo.college.EmployeeDAO"*>

<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*></property>

</bean>

</beans>

**Output:-**

****

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

**Employee.java**

**package** demo.rm;

**public class** Employee {

# private int id;

**private** String name;

**public int** getId() {

**return** id;

}

**public void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public void** setName(String name) {

**this**.name = name;

}

}

**Config.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=[*"http://www.springframework.org/schema/beans"*](http://www.springframework.org/schema/beans)xmlns:xsi=[*"http://www.w3.org/2001/XMLSchema-instance"*](http://www.w3.org/2001/XMLSchema-instance)

xsi:schemaLocation=[*"http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd) *"*>

<bean id=*"ds"*

## class="org.springframework.jdbc.datasource.DriverManagerDataSour ce">

<property name=*"driverClassName"*

value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"*

value=*"jdbc:mysql://localhost:3306/college"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"root"*></property>

</bean>

<bean id=*"jdbcTemplate"*

## class="org.springframework.jdbc.core.JdbcTemplate">

<property name=*"dataSource"* ref=*"ds"*></property>

</bean>

<bean id = *"employeeDAO"* class =*"demo.rm.EmployeeDAO"*>

<property name = *"jdbcTemplate"* ref = *"jdbcTemplate"*></property>

</bean>

</beans>

**EmployeeDAO**:

**package** demo.rm;

**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** EmployeeDAO {

**private** JdbcTemplate jdbcTemplate;

**public void** setJdbcTemplate(JdbcTemplate jdbcTemplate) {

**this**.jdbcTemplate = jdbcTemplate;

}

**public** List<Employee> findall(){

String query="select \* from employee";

List<Employee> employees =

jdbcTemplate.query(query,**new** RowMapper<Employee>() { @Override

**public** Employee mapRow(ResultSet arg0, **int** arg1)

**throws** SQLException {

// **TODO** Auto-generated method stub Employee e= **new** Employee(); e.setId(arg0.getInt(1)); e.setName(arg0.getString(2));

**return** e;

}

});

**return** employees;

}

}

**MainApp:**

**package** demo.rm;

**import** java.util.List;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationConte xt;

**public class** MainApp {

**public static void** main(String[] args) {

// **TODO** Auto-generated method stub ApplicationContext con= **new**

ClassPathXmlApplicationContext("config.xml");

EmployeeDAO

e\_dao=con.getBean("employeeDAO",EmployeeDAO.**class**);

Table:");

List<Employee> emps=e\_dao.findall();

System.***out***.println("");

System.***out***.println("Data Using RowMapper from Employee

**for**(Employee e:emps) {

System.***out***.println("Id"+e.getId()); System.***out***.println("Name"+e.getName());

}

}

}

**Output:**