**12.2 Aim: Write a program to demonstrate RowMapper interface to fetch the records from the database.**

**Code –**

**On pgAdmin4**

create table employee(id int,name varchar, salary int );

select \* from employee;

**Employee1.java**

**package** org.viva;

**public** **class** Employee1 {

**private** **int** id;

**private** String name;

**private** **int** salary;

**public** Employee1() {

**super**();

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

}

**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** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** Employee1(**int** id, String name, **int** salary) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

}

}

**EmployeeDao.java**

**package** org.viva;

**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** EmployeeDao() {

**super**();

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

}

**public** EmployeeDao(JdbcTemplate jdbcTemplate) {

**super**();

**this**.jdbcTemplate = jdbcTemplate;

}

**public** JdbcTemplate getJdbcTemplate() {

**return** jdbcTemplate;

}

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

**this**.jdbcTemplate = jdbcTemplate;

}

**public** List<Employee1> getAllEmployees(){

**return** jdbcTemplate.query("select \* from employee",**new** ResultSetExtractor<List<Employee1>>(){

@Override

**public** List<Employee1> extractData(ResultSet rs) **throws** SQLException,

DataAccessException {

List<Employee1> list=**new** ArrayList<Employee1>();

**while**(rs.next()){

Employee1 e=**new** Employee1();

e.setId(rs.getInt(1));

e.setName(rs.getString(2));

e.setSalary(rs.getInt(3));

list.add(e);

}

**return** list;

}

});

}

}

**EmployeeTest1.java**

**package** org.viva;

**import** java.util.List;

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

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

**public** **class** EmployeeTest1 {

**private** **static** ApplicationContext *ctx*;

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

*ctx* = **new** ClassPathXmlApplicationContext("Appctx3.xml");

EmployeeDao dao=(EmployeeDao)*ctx*.getBean("edao");

List<Employee1> list=dao.getAllEmployees();

**for**(Employee1 e:list)

System.***out***.println(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:5433/postgres" />

<property name="username" value="postgres" />

<property name="password" value="abc" />

</bean>

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

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

</bean>

<bean id="edao" class="org.viva.EmployeeDao">

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

</bean>

</beans>