

```
CREATE TABLE "EMP558"  
( "ID" NUMBER(10,0) NOT NULL ENABLE,  
  "NAME" VARCHAR2(255 CHAR),  
  "SALARY" FLOAT(126),  
  PRIMARY KEY ("ID") ENABLE  
)
```

```
package com;
```

```
public class Employee {  
    private int id;  
    private String name;  
    private float salary;  
    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 getSalary() {  
        return salary;  
    }  
}
```

```
public void setAddress(String Salary) {  
    this.salary = salary;  
}
```

```
<?xml version='1.0' encoding='UTF-8'?>  
<!DOCTYPE hibernate-mapping PUBLIC  
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
```

```
<hibernate-mapping>  
<class name="com.javatpoint.Employee" table="emp558">  
    <id name="id">  
        <generator class="assigned"></generator>  
    </id>  
  
    <property name="name"></property>  
    <property name="salary"></property>  
</class>  
  
</hibernate-mapping>
```

```
package com;  
  
import org.springframework.orm.hibernate3.HibernateTemplate;  
import java.util.*;  
  
public class EmployeeDao {  
    HibernateTemplate template;  
  
    public void setTemplate(HibernateTemplate template) {
```

```
        this.template = template;
    }

    //method to save employee
    public void saveEmployee(Employee e){
        template.save(e);
    }

    //method to update employee
    public void updateEmployee(Employee e){
        template.update(e);
    }

    //method to delete employee
    public void deleteEmployee(Employee e){
        template.delete(e);
    }

    //method to return one employee of given id
    public Employee getById(int id){
        Employee e=(Employee)template.get(Employee.class,id);
        return e;
    }

    //method to return all employees
    public List<Employee> getEmployees(){
        List<Employee> list=new ArrayList<Employee>();
        list=template.loadAll(Employee.class);
        return list;
    }
}
```

```
<?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="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
```

```
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"></property>
```

```
    <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"></property>
```

```
    <property name="username" value="system"></property>
```

```
    <property name="password" value="oracle"></property>
```

```
</bean>
```

```
<bean id="mySessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
```

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

```
    <property name="mappingResources">
```

```
        <list>
```

```
            <value>employee.hbm.xml</value>
```

```
        </list>
```

```
    </property>
```

```
    <property name="hibernateProperties">
```

```
        <props>
```

```
            <prop key="hibernate.dialect">org.hibernate.dialect.Oracle9Dialect</prop>
```

```

        <prop key="hibernate.hbm2ddl.auto">update</prop>
        <prop key="hibernate.show_sql">true</prop>

    </props>
</property>
</bean>

<bean id="template" class="org.springframework.orm.hibernate3.HibernateTemplate">
    <property name="sessionFactory" ref="mySessionFactory"></property>
</bean>

<bean id="d" class="com.javatpoint.EmployeeDao">
    <property name="template" ref="template"></property>
</bean>

</beans>

```

```

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;
import org.springframework.core.io.Resource;

```

```

public class InsertTest {
    public static void main(String[] args) {

```

```

        Resource r=new ClassPathResource("applicationContext.xml");

```

```
BeanFactory factory=new XmlBeanFactory(r);
```

```
EmployeeDao dao=(EmployeeDao)factory.getBean("d");
```

```
Employee e=new Employee();
```

```
e.setId(114);
```

```
e.setName("varun");
```

```
e.setSalary(50000);
```

```
dao.saveEmployee(e);
```

```
}
```

```
}
```