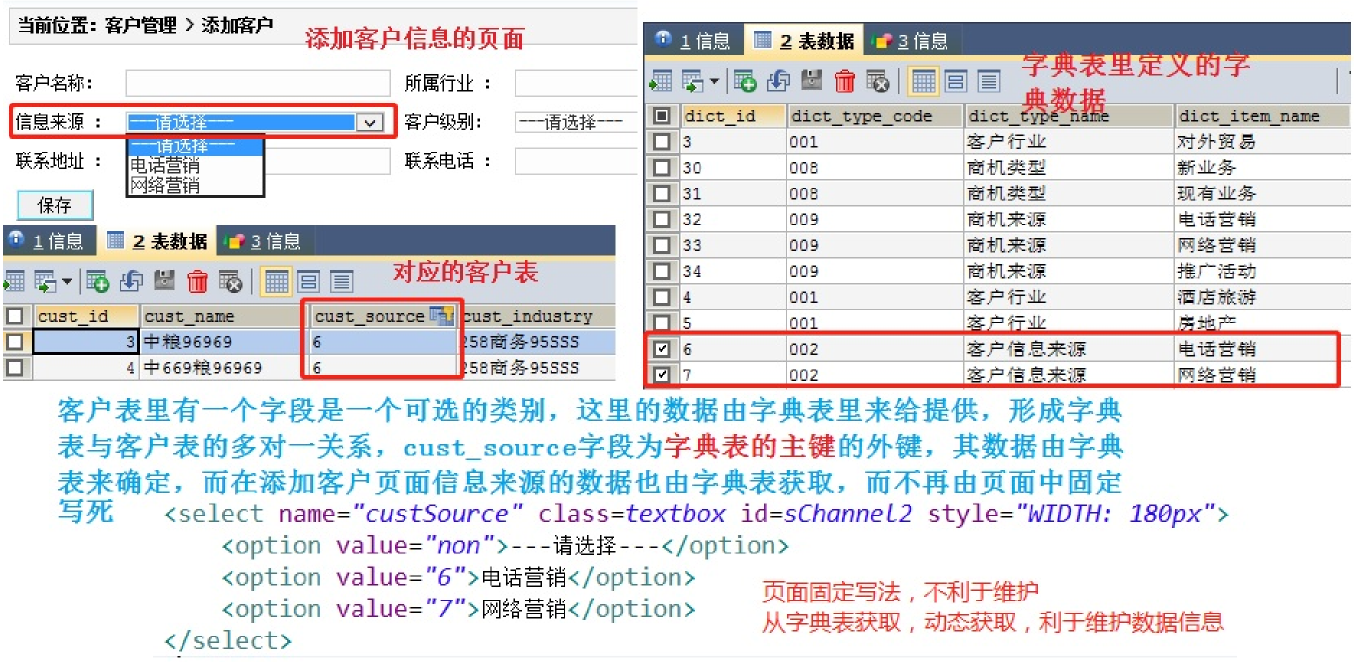
字典表：



当前位置：客户管理 > 添加客户

添加客户信息的页面

|  |  |  |  |
|---|---|---|---|
| 客户名称： |  | 所属行业 ： |  |
| 信息来源 ： | ---请选择--- | 客户级别： | ---请选择--- |
| 联系地址 ： |  | 联系电话 ： |  |

---请选择---
电话营销
网络营销

保存

对应的客户表

| □ | cust_id | cust_name | cust_source | cust_industry |
|---|---|---|---|---|
| □ | 3 | 中粮96969 | 6 | 258商务95SSS |
| □ | 4 | 中669粮96969 | 6 | 258商务95SSS |

字典表里定义的字典数据

| ■ | dict_id | dict_type_code | dict_type_name | dict_item_name |
|---|---|---|---|---|
| □ | 3 | 001 | 客户行业 | 对外贸易 |
| □ | 30 | 008 | 商机类型 | 新业务 |
| □ | 31 | 008 | 商机类型 | 现有业务 |
| □ | 32 | 009 | 商机来源 | 电话营销 |
| □ | 33 | 009 | 商机来源 | 网络营销 |
| □ | 34 | 009 | 商机来源 | 推广活动 |
| □ | 4 | 001 | 客户行业 | 酒店旅游 |
| □ | 5 | 001 | 客户行业 | 房地产 |
| ☑ | 6 | 002 | 客户信息来源 | 电话营销 |
| ☑ | 7 | 002 | 客户信息来源 | 网络营销 |

客户表里有一个字段是一个可选的类别，这里的数据由字典表里来给提供，形成字典表与客户表的多对一关系，cust_source字段为字典表的主键的外键，其数据由字典表来确定，而在添加客户页面信息来源的数据也由字典表获取，而不再由页面中固定写死

```
<select name="custSource" class=textbox id=sChannel2 style="WIDTH: 180px">
    <option value="non">---请选择---</option>
    <option value="6">电话营销</option>
    <option value="7">网络营销</option>
</select>
```

页面固定写法，不利于维护
从字典表获取，动态获取，利于维护数据信息

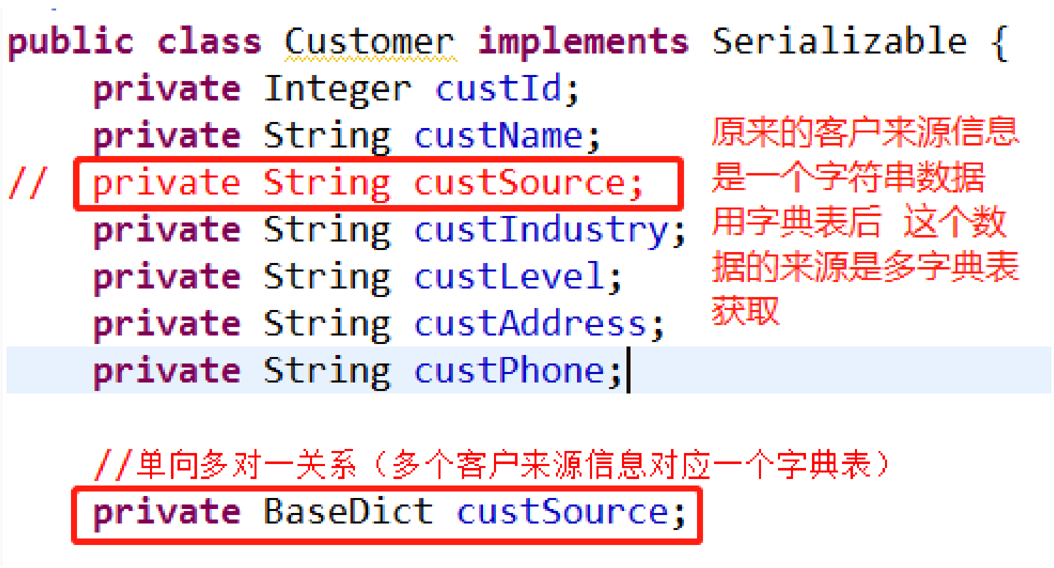
字典表实现步骤：

1、创建字典表

2、根据字典表创建实体类并创建字典表与实体类的映射配置文件.hbm.xml

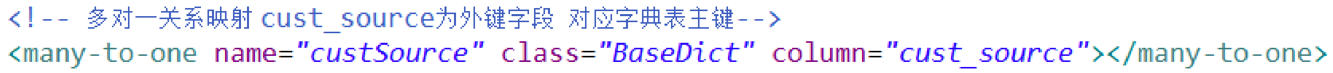
3、确定字典表主键对应的数据是原数据表的哪个字段，修改原数据表的对应关系，一般为多对一关系

原数据表某一字段对应字典表获取



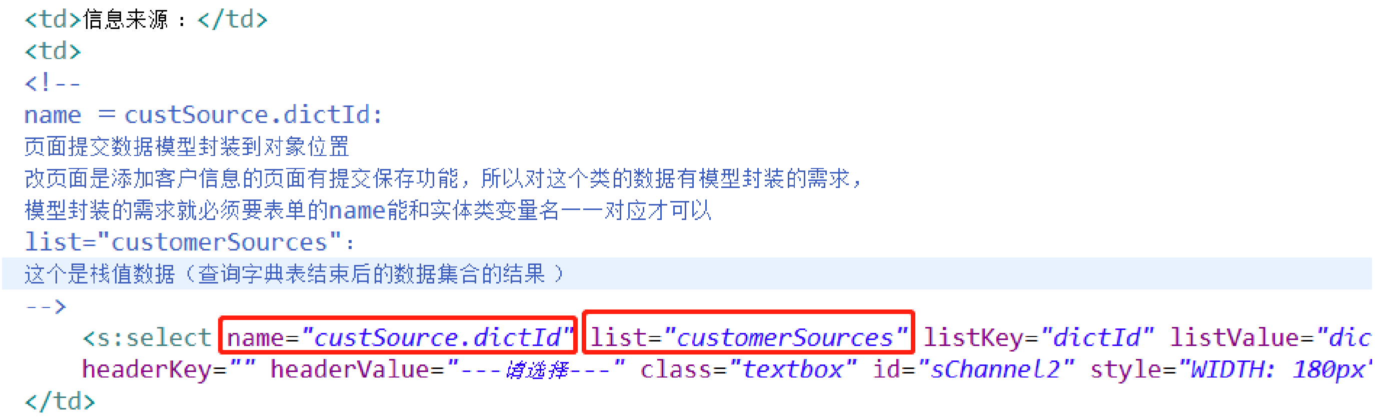
```
public class Customer implements Serializable {
    private Integer custId;
    private String custName;
//  private String custSource;
    private String custIndustry;
    private String custLevel;
    private String custAddress;
    private String custPhone;

    //单向多对一关系（多个客户来源信息对应一个字典表）
    private BaseDict custSource;
```

原来的客户来源信息是一个字符串数据 用字典表后 这个数据的来源是多字典表获取



```
<!-- 多对一关系映射 cust_source为外键字段 对应字典表主键-->
<many-to-one name="custSource" class="BaseDict" column="cust_source"></many-to-one>
```

4、修改页面数据



```
<td>信息来源：</td>
<td>
<!--
name = custSource.dictId:
页面提交数据模型封装到对象位置
改页面是添加客户信息的页面有提交保存功能，所以对这个类的数据有模型封装的需求，
模型封装的需求就必须要表单的name能和实体类变量名一一对应才可以
list="customerSources":
这个是栈值数据（查询字典表结束后的数据集合的结果 ）
-->
    <s:select name="custSource.dictId" list="customerSources" listKey="dictId" listValue="dic
    headerKey="" headerValue="---请选择---" class="textbox" id="sChannel2" style="WIDTH: 180px'
</td>
```

## html列表颜色设置：list.jsp

<TR ${vs.odd ? "style='background-color:#FFFACD'":"style='background-color:#F0F8FF'"} style="FONT-WEIGHT: normal; FONT-STYLE: normal; BACKGROUND-COLOR: white; TEXT-DECORATION: none">