#### 主表———> 外键 <———从表 关系:

外键:从表中一列的数据来源于主表的主键,我们把这一列称之为外键。外键的数据可以是null,当不为null时必须是主表主键中出现过的

主表

| 客户表(cst_customer) |           |              |  |
|-------------------|-----------|--------------|--|
| cust_id           | cust_name | cust_address |  |
| 1                 | 传智播客      | 北七家          |  |
| 2                 | 黑马程序员     | 软件园          |  |
| 3                 | 中粮商务公园    | 留仙二路         |  |
| 4                 | 华强北       | 华强北          |  |
|                   |           |              |  |
|                   |           |              |  |

| 从表                |          |            |             |  |
|-------------------|----------|------------|-------------|--|
| 联系人表(cst_linkman) |          |            |             |  |
| lkm_id            | 1km_name | lkm_gender | customer id |  |
| 1                 | 联系人1     | male       | 1           |  |
| 2                 | 联系人 2    | female     | 1           |  |
| 3                 | 联系人 3    | male       | 2           |  |
| 4                 | 联系人 4    | female     | 2           |  |
| 5                 | 联系人 5    | male       | 3           |  |
| 6                 | 联系人 6    | female     | 3           |  |
| 7                 | 联系人 7    | male       | 4           |  |
| 8                 | 联系人 8    | female     | 4           |  |

外键

主表为客户表保存所有的客户信息一个客户可以有多个联系人

从表保存联系人多个联系人属于一个客户

#### 多表关系类型:

- 1、一对多(实际使用最多)
- 2、多对一(实际使用最多)
- 3、多对多(实际使用较多)
- 4、一对一(实际基本不使用)

### 单向多对一实体类的关系映射

```
* 对应数据库表的实体JAVABean类 实现序列化 并生成get/set方法
* 主表(客户)信息
                                                        * 联系人管理Bean信息
                                                        * 从表 客户的联系人信息
                                                        */
public class Customer implements Serializable {
   private Integer cusId;
                                                       public class Linkman implements Serializable {
   private String custName;
   private String custSource;
                                                           private Long lkmId;
   private String custIndustry;
                                                           private String lkmName;
   private String custLevel;
                                                           private String lkmGender;
   private String custAddress;
                                                           private String lkmPhone;
   private String custPhone;
                                                           private String lkmMobile;
                                                           private String lkmEmail;
                                                           private String lkmPosition;
                        单向多对一映射的实体类关系
                                                           private String lkmMemo;
                                                           // 单向多对一关系映射:多个联系人对应一个客户
                                                           private Customer customer;
```

```
Customer.hbm.xml
        <!-- 映射实体类中属性与表中非主键列名 -->
           - 多对一关系映射
                                                          cproperty name="custName" column="cust name">
           涉及的标签: many-to-one
                                                          cproperty name="custSource" column="cust_source"
                                                          cproperty name="custIndustry" column="cust_in
                  作用:用于多对一关系映射
                                                          column="cust level" column="cust level
                                                          cproperty name="custAddress" column="cust add.
                     name: 指定实体类中引用对象的名称。get
                                                          cproperty name="custPhone" column="cust phone
                      column:指定外键字段的名称
                                                       </class>
                     class:指定的引用对象的实体类名称
                                                    </hibernate-mapping>
         <many-to-one name="customer"</pre>
         class="Customer"
                                                       单向多对一映射文件配置:
         column="lkm_cust_id" >
         </many-to-one>
                                                       多的一方配置映射关系
                                                       一的一方不变
 </hibernate-mapping>
```

## 单向一对多实体类的关系映射

```
* 对应数据库表的实体JAVABean类 实现序列化 并生成get/set方法
                                                                                      单向一对多实体类映射关系
                                                                  * 联系人管理Bean信息
 * 主表(客户)信息
                                                                  * 从表 客户的联系人信息
public class Customer implements Serializable {
    private Integer cusId;
                                                                 public class Linkman implements Serializable {
    private String custName;
                                                                     private Long lkmId;
    private String custSource;
    private String custIndustry;
                                                                     private String lkmName;
                                                                     private String lkmGender;
    private String custLevel;
                                                                     private String lkmPhone;
    private String custAddress;
    private String custPhone;
                                                                     private String lkmMobile;
                                                                     private String lkmEmail;
    //单向一对多关系映射:一个客户可以包含多个联系人
                                                                     private String lkmPosition;
    private Set<Linkman> linkmans =
                                                                     private String lkmMemo;
            new HashSet<Linkman>(0);

<p
                                                          cproperty name="custIndustry" column="cust_indus"
       <class name="Linkman" table="
                                                          cproperty name="custLevel" column="cust level">
            <id name="lkmId" column=
                                                          cproperty name="custAddress" column="cust_addres
                 <generator class="nai_</pre>
                                                          cproperty name="custPhone" column="cust phone">
            </id>
            <!-- 映射实体类中属性与表中非主
                                                          <!-- 一对多关系映射 涉及的标签: set:作用:用于映射set集合属
            cproperty name="lkmName"
                                                               table: 指定的是集合元素所对应的数据库表。在一对多映射时间
            property name="lkmGender
                                                               属性: class: 指定集合元素对应的实体类 -->
            property name="lkmPhone"

                                                          <set name="linkmans" table="cst_linkman" cascade</pre>
            property name="lkmMobile"
                                                               <key column="lkm_cust_id"></key>
            property name="lkmEmail
                                                               <one-to-many class="Linkman" />
            cproperty name="lkmPosit"
            cproperty name="lkmMemo"
                                                      </class>
       </class>
                                                                                单向一对多配置映射,
                                                 </hibernate-mapping>
  </hibernate-mapping>
                                                                                一的一方配置,多的一方不变
```

# 双向一对多或多对一(主从都配置)

```
* 对应数据库表的实体JAVABean类 实现序列化 并生成get
  主表(客户)信息
                                                * 联系人管理Bean信息
 */
                                                * 从表 客户的联系人信息
public class Customer implements Serializat
                                             public class Linkman implements Serializable
   private Integer cusId;
   private String custName;
   private String custSource;
                                                   private Long lkmId;
   private String custIndustry;
                                                   private String lkmName;
                                                   private String lkmGender;
   private String custLevel;
   private String custAddress;
                                                   private String lkmPhone;
   private String custPhone;
                                                   private String lkmMobile;
                                                   private String lkmEmail;
   //双向一对多关系映射:一个客户可以包含多个联系人
                                                   private String lkmPosition;
   private Set<Linkman> linkmans =
                                                   private String lkmMemo;
           new HashSet<Linkman>(0);
                                                   // 双向多对一关系映射:多个联系人对应一个客户
                                                   private Customer customer;
                                               X Linkman.hbm.xml ⋈
cproperty name= custlevel column= c
                                                                  人示坏加
                                                             涉及的标签: many-to-one
         property name="custAddress" column=
                                                                    作用:用于多对一关系映射
         cproperty name="custPhone" column="c
                                                                    属性:
                                                                        name: 指定实体类中引用对象
         <!-- 一对多关系映射 涉及的标签: set:作用:用于
                                                                        column:指定外键字段的名和
             table: 指定的是集合元素所对应的数据库表。
                                                                        class:指定的引用对象的实
             属性: class: 指定集合元素对应的实体类 -->
                                                          <many-to-one name="customer"</pre>
         <set name="linkmans" table="cst_link</pre>
                                                          class="Customer"
             <key column="lkm cust id"></key>
                                                          column="lkm cust id" >
             <one-to-many class="Linkman" />
                                                          </many-to-one>
         </set>
                                                     </class>
      </class>
                                                 </hibernate-mapping>
  </hibernate-mapping>
```

## 多表映射具体的配置内容: 重要

一对多:

<!-- 一对多关系映射 涉及的标签:

set:作用:用于映射set集合属性。

属性:

name: 指定集合属性的名称。找get/set方法后面的部分

table: 指定的是集合元素所对应的数据库表。在一对多映射

时可以不写。

key:作用:用于映射外键。

属性: column: 指定外键的名称。

one-to-many:作用:指定当前实体和集合元素之间的关系是一对

```
属性: class: 指定集合元素对应的实体类 -->
       <set name="linkmans"</pre>
           table="cst linkman" >
           <key column="lkm_cust_id"></key>
           <one-to-many class="Linkman" />
       </set>
  <!-- 一对多关系映射 涉及的标签:
   set:作用:用于映射set集合属性。
  属性:
      name: 指定集合属性的名称。找get/set方法后面的部分
      table: 指定的是集合元素所对应的数据库表。在一对多映射时可以不写。
  key:作用:用于映射外键。
      属性: column: 指定外键的名称。
  one-to-many:作用:指定当前实体和集合元素之间的关系是一对多。
      属性: class: 指定集合元素对应的实体类 -->
  <set name="linkmans"
      table="cst linkman" >
      <key column="lkm cust id"></key>
      <one-to-many class="Linkman" />
  </set>
多对一:
<!-- 多对一关系映射
           涉及的标签: many-to-one
                  作用:用于多对一关系映射
                  属性:
                      name: 指定实体类中引用对象的名称。get/set
方法后面的部分。
                      column:指定外键字段的名称
                      class:指定的引用对象的实体类名称
```

```
<many-to-one
       name="customer"
       class="Customer"
       column="lkm cust id" >
       </many-to-one>
<!-- 多对一关系映射
   涉及的标签: many-to-one
          作用:用于多对一关系映射
          属性:
              name: 指定实体类中引用对象的名称。get/set方法后面的部分。
              column:指定外键字段的名称
              class:指定的引用对象的实体类名称 -->
<many-to-one
name="customer"
class="Customer"
column="lkm cust id" >
</many-to-one>
```

多表映射的其它配置: 作用于映射文件上的配置

- 1、更新时的联级保存更新配置 cascade="save-update"
- 2、双向保存时的主表放弃关联关系维护配置,只能在set标签上 inverse="true"

#### 在Hibernate中实现

单向一对多, 多的一方(从表映射不配置映射关系)

单向多对一,一的一方(主表映射不配置映射关系)

双向一对多或多对一,主表 从表映射都必须配置映射关系