###### 关联查询

当查询内容涉及到具有关联关系的多个表时，需要使用关联查询。根据表与表之间的关联关系的不同，关联查询分为四种：

1. 一对一关联查询
2. 一对多关联查询
3. 多对一关联查询
4. 多对多关联查询

一对一关联查询的解决方案和多对一解决方案是相同的。

外键定义在多方表。

如果一个表里面有外键，它肯定充当着多方表。

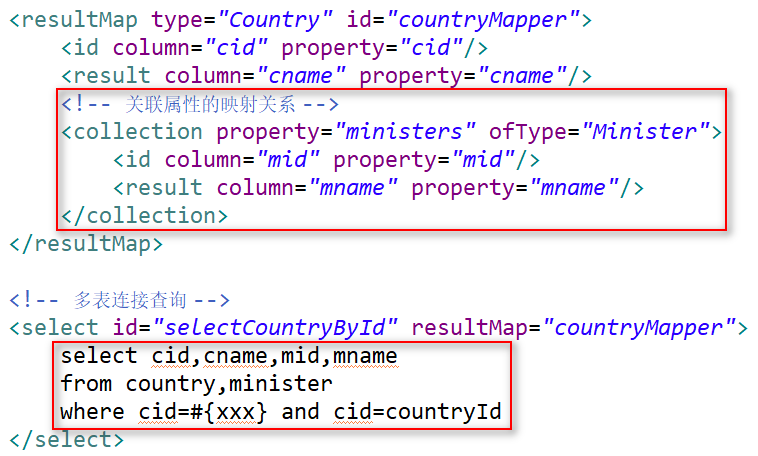
###### 一对多关联查询

这里的一对多关联查询是指，在查询一方对象的时候，同时将其所关联的多方对象也都查询出来。

以下以国家Country与部长Minister间的一对多关系进行演示。

定义映射文件：

* 多表连接查询方式



注意，此时即使字段名与属性名相同，在<resultMap/>中也要写出他们的映射关系，因为框架是依据这人<resultMap/>封装对象的。

另外，在映射文件中使用<collection/>标签体现出两个实体对象间的关联关系。其两个属性的意义为：  
 property：指定关联属性，即Country类中的集合属性

ofType：集合属性的泛型类型

* 多表单独查询方式（使用较多，这种方式可以使用延迟加载）

多表连接查询方式是将多张表进行连接，连为一张表后进行查询。其查询的本质是一张表。而多表单查询方式是多张表各自查询各自的内容，需要多张表的联合数据，则将主表的查询结果联合其他表的查询结果，封装为一个对象。

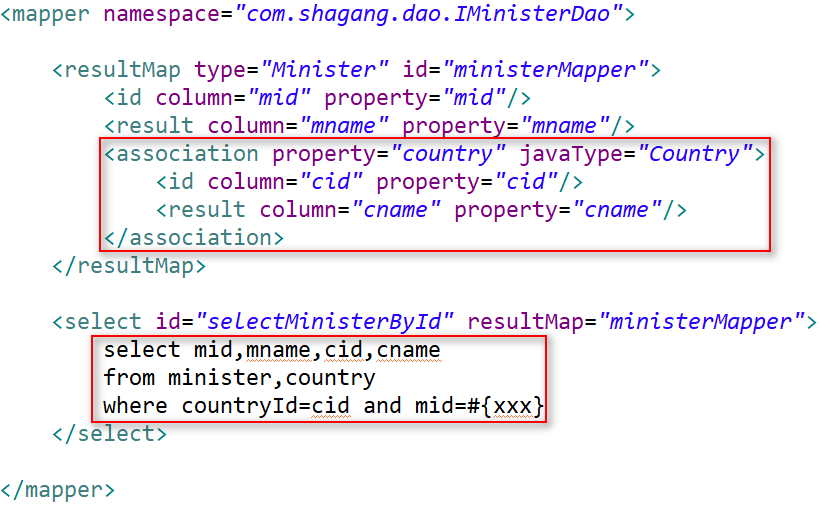
当然，这多个查询时可以跨越多个映射文件的，即是可以跨越多个namespace的。在使用其他namespace的查询时，添加上其所在的namespace即可。



关联属性<collection/>的数据来自于另一个查询<selectMinisterByCountry/>。而该查询<selectMinisterByCountry/>的动态参数countryId=#{ooo}的值来自于查询<selectCountryById/>的查询结果字段cid。

###### 多对一关联查询

* 多表连接查询方式



注意：在映射文件中使用<association/>标签体现出两个实体对象间的关联关系。

* 多表单独查询方式



###### 自关联查询

所谓自关联查询是指，自己即充当一方，又充当多方，是1：n或n：1的变型。例如，对于新闻栏目NewsColumn，可以充当一方，也可以充当多方，即子栏目。而反映到DB表中，，只有一张表，这表中具有一个外键，用于表示该栏目的父栏目。一级栏目没有父栏目，所以可以将其外键值设置为0，而子栏目则具有外键值。

为了便于理解，将自关联分为两种情况来讲解。一种是当做1：n讲解，即当前类作为一方，其包含多方的集合域属性。一种是当做n：1讲解，即当前类作为多方，其包含一方的域属性。

下面以新闻栏目为例进行讲解。由于Column是DBMS中的关键字，为了避免误解，将新闻栏目实体类定义为NewsLabel。

1. **自关联的DB表**



1. **以一对多方式处理**

以一对多方式处理，即一方可以看到多方。该处理方式的应用场景比较多，例如在页面上点击父栏目，显示出其子栏目。再如，将鼠标定位在窗口中的某菜单项上会显示其所有子菜单项目等。

根据查询需求的不同，又可以分为两种情况：一种是查询出指定栏目的所有子孙栏目，一种是查询出指定栏目及其所有的子孙栏目。

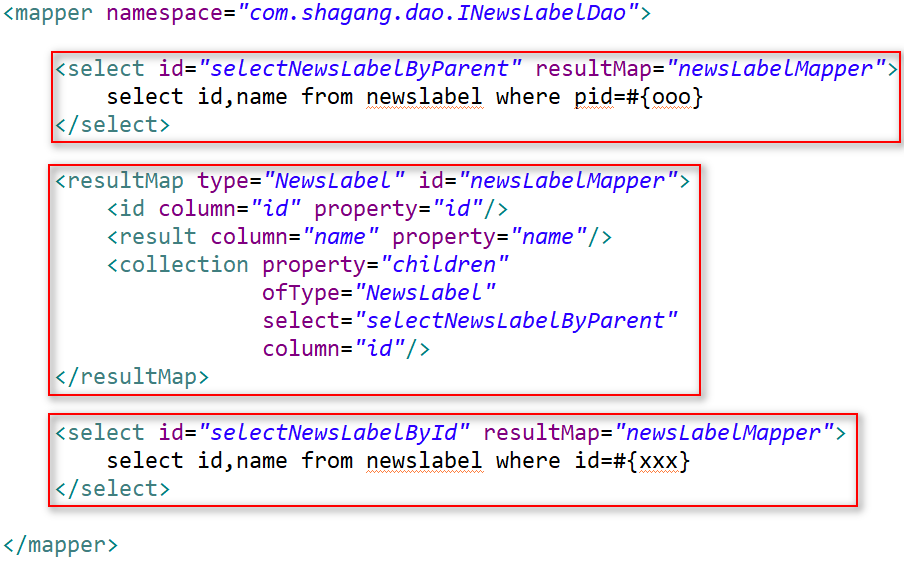
1. **查询指定栏目的所有子孙栏目**

这里通过select语句的递归调用实现查询所有下级栏目功能。查询结果的集合数据<collection/>来自于递归调用的selectChildrenByParent查询。与第一次进行查询不同的是，第一次的pid动态参数来自于调用方法传递来的实参，而<collection/>中查询语句的动态pid动态参数值来自于上一次的查询结果的id值。



1. **查询指定栏目及其所有子孙栏目**

这里的查询结果，既要包含指定id的当前栏目，还要包含其所有辈分的孙子栏目。即给出的id实际为当前要查询的栏目的id。



1. **以多对一方式处理**

以多对一方式处理，即多方可以看到一方。该处理方式的应用场景，例如在网页上显示当前页面的站内位置。

###### 多对多关联查询

什么是多对多关联关系？一个学生可以选多门课程，而一门课程可以由多个学生选。这就是典型的多对多关系。所以，所谓多对多关系，其实就是由两个互反的一对多关系组成。一般情况下多对多关系都会通过一个中间表来建立，例如选课表。

注：在定义双向关联（双方均可看到对方的关联关系）的实体的toString()方法时，只让一方的toString()方法中可以输出对方，不要让双方均可输出对方。否则将回出现输出时的递归现象，程序报错。

定义数据库表



