



# 黑马程序员

## 《Java EE 企业级应用开发教程 （SSM）（第2版）》

### 教学设计

课程名称： Java EE 企业级应用开发教程

授课年级： XXXX 年级

授课学期： 第 X 学期

教师姓名： 某某老师

2021年6月



课题名称	第4章 MyBatis 的关联映射和缓存机制	计划课时	4 课时
教学引入	对数据库的操作常常会涉及到多张表，针对多表之间的操作，MyBatis 提供了关联映射，通过关联映射可以很好地处理表与表、对象与对象之间的关联关系。此外，在实际开发中经常需要合理地利用 MyBatis 缓存来加快数据库查询，进而有效地提升数据库性能。本章将针对 MyBatis 的关联映射和缓存机制进行详细讲解。		
教学目标	<ul style="list-style-type: none"> <li>● 使学生了解数据表之间的三种关联关系</li> <li>● 使学生了解对象之间的三种关联关系</li> <li>● 使学生熟悉关联关系中的嵌套查询和嵌套结果</li> <li>● 使学生掌握一对一关联映射</li> <li>● 使学生掌握一对多关联映射</li> <li>● 使学生掌握多对多关联映射</li> <li>● 使学生熟悉 MyBatis 的缓存机制</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● 一对一查询</li> <li>● 一对多查询</li> <li>● 多对多查询</li> </ul>		
教学难点	<ul style="list-style-type: none"> <li>● 一对多查询</li> <li>● 多对多查询</li> <li>● 二级缓存</li> </ul>		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
教学过程	<p style="text-align: center;"><b>第一课时</b> <b>（关联映射概述、一对一查询、一对多查询）</b></p> <p>一、创设情景，导入新课</p> <p>事物之间都存在着某种联系。在程序中，数据库的表之间也存在某种联系，我们称为关联映射。本节课，将对关联映射概述、一对一查询、一对多查询等内容进行详细讲解。</p> <p>二、新课讲解</p> <p><b>知识点 1-关联映射概述</b></p> <p>教师通过多媒体演示 PPT 内容讲解关联映射概述。</p> <p>在关系型数据库中，表与表之间存在着三种关联映射关系，分别为一对一、一对多和多对多。</p> <p>这三种关联映射关系的具体说明如下。</p> <ul style="list-style-type: none"> <li>● 一对一：一个数据表中的一条记录最多可以和另一个数据表中的一条记录相关。</li> <li>● 一对多：主键数据表中的一条记录可以和另外一个数据表的多条记录相关。但另外一个数据表中的记录只能与主键数据表中的某一条记录相关。</li> <li>● 多对多：一个数据表中的一条记录可以与另外一个数据表任意数量的记录相关，另外一个数据表中的一条记录也可以与本数据表中任意数</li> </ul>		



量的记录相关。

数据表之间的关系实质上描述的是数据之间的关系，除了数据表，在 Java 中，还可以通过对象也描述数据之间的关系。通过 Java 对象描述数据之间的关系，其实就是使对象的属性与另一个对象的属性相互关联。

三种 Java 对象关联映射关系的描述如下：

- 一对一：就是在本类中定义与之关联的类的对象作为属性。
- 一对多：就是一个 A 类对象对应多个 B 类对象的情况。
- 多对多：在两个相互关联的类中，都可以定义多个与之关联的类的对象。

### 知识点 2—一对一查询

教师讲解一对一查询。

在 MyBatis 中，通过 `<association>` 元素来处理一对一关联关系。`<association>` 元素提供了一系列属性用于维护数据表之间的关系。

`<association>` 元素是 `<resultMap>` 元素的子元素，它有两种配置方式，嵌套查询和嵌套结果，下面对这两种配置方式分别进行介绍。

#### 1. 嵌套查询方式

嵌套查询是指通过执行另外一条 SQL 映射语句来返回预期的复杂类型。

#### 2. 嵌套结果方式

嵌套结果是使用嵌套结果映射来处理重复的联合结果的子集。

接下来就以个人和身份证之间的一对一关联关系为例，对 MyBatis 中一对一关联关系的处理进行详细讲解。案例具体实现步骤如下。

##### (1) 创建数据表。

在 mybatis 数据库中分别创建名为 `tb_idcard` 的身份证数据表和名称为 `tb_person` 的个人数据表，同时预先插入两条数据。

##### (2) 持久化类 `IdCard`。

在项目的 `com.itheima.pojo` 包下创建持久化类 `IdCard`，用于封装身份证属性。

##### (3) 持久化类 `Person`。

在项目的 `com.itheima.pojo` 包下创建持久化类 `Person`，用于封装个人属性。

##### (4) 创建身份证映射文件。

在 `com.itheima.mapper` 包中，创建身份证映射文件 `IdCardMapper.xml`，并在映射文件中编写一对一关联映射查询的配置信息。

##### (5) 创建人员映射文件。

在 `com.itheima.mapper` 包中，创建人员映射文件 `PersonMapper.xml`，并在映射文件中编写一对一关联映射查询的配置信息。

##### (6) 编写配置文件。

在核心配置文件 `mybatis-config.xml` 中，引入 `IdCardMapper.xml` 和 `PersonMapper.xml` 映射文件，并为 `com.itheima.pojo` 包下的所有实体类定义别名。

##### (7) 编写测试类。

为了验证上述配置，可以在测试类 `MyBatisTest` 中，编写测试方法 `findPersonByIdTest()`。



虽然使用嵌套查询的方式比较简单，但是 MyBatis 嵌套查询的方式要执行多条 SQL 语句，这对于大型数据集合和列表展示不是很好，因为这样可能会导致成百上千条关联的 SQL 语句被执行，从而极大地消耗数据库性能并且会降低查询效率，这并不是开发人员所期望的。为此，我们可以使用 MyBatis 提供的嵌套结果方式进行关联查询。

多学一招：MyBatis 延迟加载的配置

在使用 MyBatis 嵌套查询方式进行 MyBatis 关联映射查询时，使用 MyBatis 的延迟加载在一定程度上可以降低运行消耗并提高查询效率。MyBatis 默认没有开启延迟加载，需要在核心配置文件 mybatis-config.xml 中的 <settings> 元素内进行配置。

### 知识点 3-一对多查询

教师讲解一对多查询。

在 MyBatis 中，通过 <collection> 元素来处理一对多关联关系。<collection> 元素的属性大部分与 <association> 元素相同，但其还包含一个特殊属性 ofType。ofType 属性与 javaType 属性对应，它用于指定实体类对象中集合属性所包含的元素的类型。

同 <association> 元素一样，<collection> 元素也是 <resultMap> 元素的子元素，<collection> 元素也有嵌套查询和嵌套结果两种配置方式。

接下来以用户和订单之间的一对多关联关系为例，详细讲解如何在 MyBatis 中处理一对多关联关系，具体步骤如下。

(1) 创建数据表。

在名为 mybatis 的数据库中，创建两个数据表，分别为 tb\_user（用户数据表）和 tb\_orders（订单表），同时在表中预先插入几条测试数据。

(2) 创建持久化类 Orders。

在 com.itheima.pojo 包中，创建持久化类 Orders，并在类中定义订单 id 和订单编号等属性。

(3) 创建持久化类 Users。

在 com.itheima.pojo 包中，创建持久化类 Users，并在类中定义用户编号、用户姓名、用户地址以及用户关联的订单等属性。

(4) 创建映射文件。

在 com.itheima.mapper 包中，创建用户实体映射文件 UsersMapper.xml，并在文件中编写一对多关联映射查询的配置。

(5) 编写配置文件。

在核心配置文件 mybatis-config.xml 中，引入 UsersMapper.xml，将 UsersMapper.xml 映射文件加载到程序中。

(6) 编写测试类。

在测试类 MyBatisTest 中，编写测试方法 findUserTest()。

需要注意的是，上述案例从用户的角度出发，用户与订单之间是一对多的关联关系，但如果从单个订单的角度出发，一个订单只能属于一个用户，即一对一的关联关系。

### 三、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。



#### 四、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

#### 第二课时

##### (多对多查询、一级缓存、二级缓存、案例：商品的类别)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接导入的方式导入新课

掌握了关联映射的基本知识以及一对一查询、一对多查询，接下来继续学习复杂的多对多查询以及 MyBatis 的缓存机制等。

#### 三、新课讲解

##### 知识点 1-多对多查询

教师讲解多对多查询。

在实际项目开发中，多对多的关联关系也是非常常见的。以订单和商品为例，一个订单可以包含多种商品，而一种商品又可以属于多个订单，订单和商品就属于多对多的关联关系。

在数据库中，多对多的关联关系通常使用一个中间表来维护，中间表中的订单 id 作为外键关联订单表的 id，中间表中的商品 id 作为外键关联商品表的 id。

下面以订单表与商品表之间的多对多关系为例来讲解如何使用 MyBatis 处理多对多的关系，具体实现步骤如下。

##### (1) 创建数据表。

在名为 mybatis 的数据库中创建名称为 tb\_product 的商品表和名称为 tb\_ordersitem 的中间表，同时在表中预先插入几条数据。

##### (2) 创建持久化类 Product。

在 com.itheima.pojo 包中，创建持久化类 Product，并在类中定义商品 id、商品名称、商品单价等属性，以及与订单关联的属性。

##### (3) 编写商品持久化类。

在商品持久化类中，除了需要添加订单的集合属性外，还需要在订单持久化类 (Orders.java) 中增加商品集合的属性及其对应的 getter/setter 方法，同时为了方便查看输出结果，需要重写 toString() 方法。

##### (4) 创建订单实体映射文件。

在 com.itheima.mapper 包中，创建订单实体映射文件 OrdersMapper.xml，用于编写订单信息的查询 SQL 语句，并在映射文件中编写多对多关联映射查询的配置信息。

##### (5) 创建商品实体映射文件。

在 com.itheima.mapper 包中，创建商品实体映射文件 ProductMapper.xml，用于编写订单与商品信息的关联查询 SQL 语句。

##### (6) 编写配置文件。

将新创建的映射文件 OrdersMapper.xml 和 ProductMapper.xml 的文件路径配置到核心配置文件 mybatis-config.xml 中。

##### (7) 编写测试类。

在测试类 MyBatisTest 中，编写多对多关联查询的测试方法



findOrdersTest()。

除了使用嵌套查询的方式查询订单及其关联的商品信息外，还可以使用嵌套结果的方式进行查询。

### 知识点 2-一级缓存

教师讲解一级缓存。

MyBatis 的一级缓存是 SqlSession 级别的缓存。如果同一个 SqlSession 对象多次执行完全相同的 SQL 语句时，在第一次执行完成后，MyBatis 会将查询结果写入到一级缓存中，此后，如果程序没有执行插入、更新、删除操作，当第二次执行相同的查询语句时，MyBatis 会直接读取一级缓存中的数据，而不用再去数据库查询，从而提高了数据库的查询效率。

接下来，通过一个案例来对 MyBatis 一级缓存的应用进行详细讲解，该案例要求根据图书 id 查询图书信息。案例具体步骤如下。

(1) 创建数据表。

在 mybatis 数据库中创建名为 tb\_book 的数据表，同时预先插入几条测试数据。

(2) 创建持久化类 Book。

在项目的 com.itheima.pojo 包下创建持久化类 Book，在 Book 类中定义图书 id、图书名称、图书价格、图书作者属性，以及属性对应的 getter/setter 方法。

(3) 创建映射文件。

在 com.itheima.mapper 包中，创建图书映射文件 BookMapper.xml，并在该文件中编写根据图书 id 查询图书信息的 SQL 语句。

(4) 编写配置文件。

在核心配置文件 mybatis-config.xml 中的 <mappers> 标签下，引入 BookMapper.xml 映射文件。

(5) 引入相关依赖。

由于需要通过 log4j 日志组件查看一级缓存的工作状态，因此需要在 pom.xml 中引入 log4j 的相关依赖。

(6) 创建 properties 文件。

在工程的 src/main/resources 目录下创建 log4j.properties 文件，用于配置 MyBatis 和控制台的日志。

(7) 编写测试方法。

为了验证上述配置，可以在测试类 MyBatisTest 中，编写测试方法 findBookByIdTest1()。

### 知识点 3-二级缓存

教师讲解二级缓存。

相同的 Mapper 类，相同的 SQL 语句，如果 SqlSession 不同，则两个 SqlSession 查询数据库时，会查询数据库两次，这样也会降低数据库的查询效率。为了解决这个问题，就需要用到 MyBatis 的二级缓存。MyBatis 的二级缓存是 Mapper 级别的缓存，与一级缓存相比，二级缓存的范围更大，多个 SqlSession 可以共用二级缓存，并且二级缓存可以自定义缓存资源。

在 MyBatis 中，一个 Mapper.xml 文件通常称为一个 Mapper，MyBatis 以





namespace 区分 Mapper，如果多个 SqlSession 对象使用同一个 Mapper 的相同查询语句去操作数据库，在第一个 SqlSession 对象执行完后，MyBatis 会将查询结果写入二级缓存，此后，如果程序没有执行插入、更新、删除操作，当第二个 SqlSession 对象执行相同的查询语句时，MyBatis 会直接读取二级缓存中的数据。

与 MyBatis 的一级缓存不同的是，MyBatis 的二级缓存需要手动开启，开启二级缓存通常要完成以下两个步骤。

#### 1. 开启二级缓存的全局配置

使用二级缓存前，需要在 MyBatis 的核心配置 mybatis-config.xml 文件中通过<settings>元素开启二级缓存的全局配置。

#### 2. 开启当前 Mapper 的 namespace 下的二级缓存

开启当前 Mapper 的 namespace 下的二级缓存，可以通过 MyBatis 映射文件中的<cache>元素来完成。

接下来通过一个案例演示 MyBatis 二级缓存的应用，该案例仍旧根据 id 查询图书信息，案例具体步骤如下。

##### (1) 修改映射文件。

修改映射文件 BookMapper.xml，在映射文件的<mapper>元素下追加编写<cache>元素开启当前 Mapper 的 namespace 的二级缓存。

##### (2) 编写测试文件。

为了验证上述配置，可以在测试类 MyBatisTest 中，编写测试方法 findByIdTest1()。

在实际开发中，经常会遇到多个 SqlSession 在同一个 Mapper 中执行操作，例如，SqlSession1 执行查询操作，SqlSession2 执行插入、更新、删除操作，SqlSession3 又执行和 SqlSession1 相同的查询操作。当 SqlSession1 执行查询操作时，程序会将查询结果写入 MyBatis 二级缓存，当 SqlSession2 对数据库执行了插入、更新、删除操作后，MyBatis 会清空二级缓存中的内容，以防止程序误读。当 SqlSession3 执行和 SqlSession1 相同的查询操作时，MyBatis 会重新访问数据库。

多学一招：Cache Hit Ratio（缓存命中率）

### 知识点 4-案例：商品的类别

教师讲解案例：商品的类别。

案例目标：

现有一个商品表 product 和一个商品类别表 category，其中，商品类别表 category 和商品表 product 是一对多的关系。

商品表 product 的详情如下表所示。

商品编号 (id)	商品名称 (goodsname)	商品详情 (price)	商品类别 (typeid)
1	电视机	5000	1
2	冰箱	4000	2
3	空调	3000	2
4	洗衣机	2000	2

商品类别表 category 详情如下表所示。



商品类别编号 (id)	商品类别名称 (typename)
1	黑色家电
2	白色家电

本案例具体要求如下。

根据表 1 和表 2 在数据库分别创建一个商品表 product 和一个商品类别表 category，并通过 MyBatis 查询商品类别为白色家电的商品的所有信息。

实现步骤如下。

#### 1. 项目搭建

创建一个名称为 mybatis-demo04 的项目，项目的具体搭建过程请参考 1.3 节。

#### 2. 数据库准备

在名为 mybatis 的数据库中，创建两个数据表，分别为 product 和 category，同时在表中预先插入几条测试数据。

#### 3. POJO 类准备

(1) 在 com.itheima.pojo 包中，创建持久化类 Category，并在类中定义商品类别的相关属性和方法。

(2) 在 com.itheima.pojo 包中，创建持久化类 Product，并在类中定义相关属性和方法。

#### 4. 编写映射文件

在 com.itheima.mapper 包中，创建商品类别实体映射文件 CategoryMapper.xml，并在文件中编写一对多关联映射查询的配置。

#### 5. 修改 mybatis-config.xml 核心配置文件

在核心配置文件 mybatis-config.xml 中，引入 CategoryMapper.xml，将 CategoryMapper.xml 映射文件加载到程序中。

#### 6. 编写测试方法

在测试类 MyBatisTest 中，编写测试方法 findCategoryTest()。

#### 四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

#### 第三、四课时（上机练习）

上机一：（考察知识点为关联映射概述、一对一查询、一对多查询、多对多查询、一级缓存、二级缓存、案例：商品的类别）

形式：单独完成

题目：

根据表 1 和表 2 在数据库分别创建一个商品表 product 和一个商品类别表 category，并通过 MyBatis 查询商品类别为白色家电的商品的所有信息。

表 1 商品表 (product)			
商品编号	商品名称	商品单价	商品类别





	(id)	(goodsname)	(price)	(typeid)
	1	电视机	5000	1
	2	冰箱	4000	2
	3	空调	3000	2
	4	洗衣机	2000	2
	表 2 商品类别表 (category)			
	商品类别编号 (id)		商品类别名称 (typename)	
	1		黑色家电	
	2		白色家电	
教学后记				