

1. MyBatis 映射器概述

元素名称	描 述	备 注
select	查询语句，最常用、最复杂的元素之一	可以自定义参数，返回结果集等
insert	插入语句	执行后返回一个整数，代表插入的行数
update	更新语句	执行后返回一个整数，代表更新的行数
delete	删除语句	执行后返回一个整数，代表删除的行数
sql	定义一部分 SQL，在多个位置被引用	例如，一张表列名，一次定义，可以在多个 SQL 语句中使用
resultMap	用来描述从数据库结果集中来加载对象，是最复杂、最强大的元素	提供映射规则

1.1 <select>元素

1.1.1 <select>元素示例

```
<select id="searchBookByTitle" resultType="Tbbook" parameterType="String">
    select * from tbbook where title=#{title}
</select>
```

上述示例代码中，id 的值是唯一标识符，它接收一个 String 类型的参数，返回一个 Tbbook 类型的对象，结果集自动映射到 Tbbook 属性

属性名称	描 述
id	它和 Mapper 的命名空间组合起来使用，是唯一标识符，供 MyBatis 调用
parameterType	表示传入 SQL 语句的参数类型的全限定名或别名。是个可选属性，MyBatis 能推断出具体传入语句的参数。
resultType	SQL 语句执行后返回的类型（全限定名或者别名）。如果是集合类型，返回的是集合元素的类型。返回时可以使用 resultType 或 resultMap 之一
resultMap	它是映射集的引用，与 < resultMap> 元素一起使用。返回时可以使用 resultType 或 resultMap 之一
flushCache	它的作用是在调用 SQL 语句后，是否要求 MyBatis 清空之前查询本地缓存和二级缓存。默认值为 false。如果设置为 true，则任何时候只要 SQL 语句被调用，都将清空本地缓存和二级缓存
useCache	启动二级缓存的开关。默认值为 true，表示将查询结果存入二级缓存中
timeout	用于设置超时参数，单位是秒。超时将抛出异常。
fetchSize	获取记录的总条数设定
statementType	告诉 MyBatis 使用哪个 JDBC 的 Statement 工作，取值为 STATEMENT

	(Statement) 、 PREPARED (PreparedStatement) 、 CALLABLE (CallableStatement)
resultSetType	这是针对 JDBC 的 ResultSet 接口而言，其值可设置为 FORWARD_ONLY (只允许向前访问)、SCROLL_SENSITIVE (双向滚动，但不及时更新)、SCROLL_INSENSITIVE (双向滚动，及时更新)

1.1.2 使用 Map 接口传递多个参数

参考“3.Spring Boot 数据访问-使用配置文件方式整合 MyBatis-动态 SQL.doc”完成实验

1.1.3 使用 Java Bean 传递多个参数

1.2 <insert>元素

1.2.1 示例

```
<insert id="insertBook" parameterType="Tbbook">
    Insert into tbbook(title,author,press,price) values(#{title},#{author},#{press},#{price})
</insert>
```

<insert>元素用于映射插入语句，MyBatis 执行完一条插入语句后，将返回一个整数表示其影响的行数。它的属性与<select>元素的属性大部分相同，在本节讲解它的几个特有属性。具体如下：

keyProperty: 该属性的作用是将插入或更新操作时的返回值赋值给 POJO 类的某个属性，通常会设置为主键对应的属性。如果是联合主键，可以在多个值之间用逗号隔开。

keyColumn: 该属性用于设置第几列是主键，当主键列不是表中的第一列时需要设置。如果是联合主键时，可以在多个值之间用逗号隔开。

useGeneratedKeys: 该属性将使 MyBatis 使用 JDBC 的 getGeneratedKeys()方法获取由数据库内部生产的主键，如 MySQL、SQL Server 等自动递增的字段，其默认值为 false。

1.2.2 主键（自动递增）回填

MySQL、SQL Server 等数据库的表格可以采用自动递增的字段作为主键。有时可能需要使用这个刚刚产生的主键，用以关联其他业务。

```
<!-- 添加一本书，成功后将主键值回填给 id (pojo 类的属性) -->
<insert id="insertBook" parameterType="Tbbook" keyProperty="id" useGeneratedKeys="true">
    Insert into tbbook(title,author,press,price) values(#{title},#{author},#{press},#{price})
```

```
</insert>
```

1.3 <update>元素与<delete>元素

<update>和<delete>元素比较简单，它们的属性和<insert>元素、<select>元素的属性差不多，执行后也返回一个整数，表示影响了数据库的记录行数。

```
<update id="updateBook" parameterType="Tbbook">
    update tbbook set title=#{title},author=#{author},press=#{press},price=#{price} where id=#{id}
</update>
```

```
    <delete id="deleteBook" parameterType="Integer">
        delete from tbbook where id=#{id}
    </delete>
```

2. <resultMap>元素

2.1 <resultMap>元素介绍

<resultMap>元素表示结果映射集，是 MyBatis 中最重要也是最强大的元素。主要用来定义映射规则、级联的更新以及定义类型转化器等。

2.2 <resultMap>元素案例

```
<select id="showComment" parameterType="Integer" resultMap="bookWithComment">
    select b.*,c.id c_id,c.content c_content,c.c_author from tbbook b,comment c
    where b.id=c.book_id and b.id=#{id}
</select>

<resultMap id="bookWithComment" type="Tbbook">
    <id property="id" column="id"/>
    <result property="title" column="title"/>
    <result property="author" column="author"/>
    <result property="press" column="press"/>
    <result property="price" column="price"/>
    <collection property="commentList" ofType="Comment">
        <id property="id" column="c_id"/>
        <result property="content" column="c_content"/>
        <result property="cAuthor" column="c_author"/>
    </collection>
</resultMap>
```

```
    </collection>  
</resultMap>  
</mapper>
```