1 MyBatis关联映射

数据库表关系

- 一对一：在任意一方引入对方主键作为外键
- 一对多：在“多”的一方添加“一”的一方的主键作为外键
- 多对多：产生中间关系表,引入两张表的主键作为外键,两个主键成为联合主键或使用新的字段作为主键

1.1 一对一

`association` 处理一对一关系

- `property`：指定映射到实体类对象属性，与表字段对应
- `column`：指定表中对应的字段
- `javaType`：指定映射到实体对象属性的类型
- `select`：指定引入嵌套查询的子 SQL 语句,用于关联映射中的嵌套查询
- `fetchType`：指定在关联查询时是否启用延迟加载，取值为 lazy 或 eager，先单表查询，需要时再进行多表查询，提高了性能

嵌套查询和联合查询（嵌套结果查询）

- 联合查询是几个表联合查询,只查询一次, 通过在resultMap 里面配置 association 节点配置一对一的类就可以完成；
- 嵌套查询是先查一个表，根据这个表里面的结果的 外键 id，去再另外一个表里面查询数据,也是通过 association 配置，但另外一个表的查询通过 select 属性配置。

1.2 一对多

`<collection>`

- property：指定映射到实体类对象属性,与表字段一-对应
- column：指定表中对应的字段
- ofType：指定映射到实体对象属性的类型,和association元素的java Type属性类似
- select：指定引入嵌套查询的子SQL语句,用于关联映射中的嵌套查询
- fetchType：指定在关联查询时是否启用延迟加载,取值为lazy或eager

2 延迟加载

数据与对象映射时，只有真正用到该对象时才进行映射操作。
resultMap中的association、collection具备延迟加载功能。

3 缓存

- 一级缓存：在 `SqlSession` 上的缓存，默认，POJO不需要序列化
- 二级缓存：在 `SqlSessionFactory` 上的缓存

1）延迟加载方法1
全局配置文件的 `<settings>` 元素内进行集中配置：

```
<settings>
  <!-- 打开延迟加载的开关 -->
  <setting name="lazyLoadingEnabled" value="true"/>
  <!-- 将积极加载改为按需加载 -->
  <setting name="aggressiveLazyLoading" value="false"/>
</settings>
```

2）延迟加载方法2
映射文件的 `<association>` 或 `<collection>` 元素中配置fetchType属性值：
`fetchType="lazy"`

3.1 一级缓存

失效：

- `SqlSession` 不同
- `SqlSession` 相同，查询条件不同（缓存未有数据）
- `SqlSession` 不同，两次查询间存在增删改操作（可能对数据有影响）
- `SqlSession` 不同，手动清除了一级缓存

3.2 二级缓存

全局缓存，基于 `namespace` 级别的缓存

工作原理

1. 会话A查询到一条数据I该数据存放在当前会话A的一级缓存中
2. 如果会话A关闭，则一-级缓存中的数据被保存到二级缓存中
3. 会话B查询该数据时，先从二级缓存中查找

查到的数据存放在一级缓存，会话提交或关闭后，一级缓存中的数据转移到二级缓存中