## MyBatis 延迟加载，一级缓存，二级缓存

1. **延迟加载**

resultMap中的association和collection标签具有延迟加载的功能。

延迟加载的意思是说，在关联查询时，利用延迟加载，先加载主信息。使用关联信息时再去加载关联信息。

* 1. **设置延迟加载**

需要在SqlMapConfig.xml文件中，在<settings>标签中设置下延迟加载。

lazyLoadingEnabled、aggressiveLazyLoading

|  |  |  |  |
| --- | --- | --- | --- |
| 设置项 | 描述 | 允许值 | 默认值 |
| lazyLoadingEnabled | 全局性设置懒加载。如果设为‘false’，则所有相关联的都会被初始化加载。 | true | false | false |
| aggressiveLazyLoading | 当设置为‘true’的时候，懒加载的对象可能被任何懒属性全部加载。否则，每个属性都按需加载。 | true | false | true |

|  |
| --- |
| <!-- 开启延迟加载 -->  <settings>  <!-- lazyLoadingEnabled:延迟加载启动，默认是false -->  <setting name="lazyLoadingEnabled" value="true"/>  <!-- aggressiveLazyLoading：积极的懒加载，false的话按需加载，默认是true -->  <setting name="aggressiveLazyLoading" value="false"/>    <!-- 开启二级缓存，默认是false -->  <setting name="cacheEnabled" value="true"/>  </settings> |

1. **查询缓存**

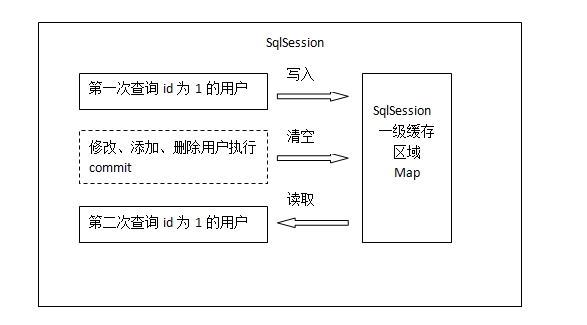
Mybatis的一级缓存是指SqlSession。一级缓存的作用域是一个SqlSession。Mybatis默认开启一级缓存。

在同一个SqlSession中，执行相同的查询SQL，第一次会去查询数据库，并写到缓存中；第二次直接从缓存中取。当执行SQL时两次查询中间发生了增删改操作，则SqlSession的缓存清空。

Mybatis的二级缓存是指mapper映射文件。二级缓存的作用域是同一个namespace下的mapper映射文件内容，多个SqlSession共享。Mybatis需要手动设置启动二级缓存。

在同一个namespace下的mapper文件中，执行相同的查询SQL，第一次会去查询数据库，并写到缓存中；第二次直接从缓存中取。当执行SQL时两次查询中间发生了增删改操作，则二级缓存清空。

1. **一级缓存原理**

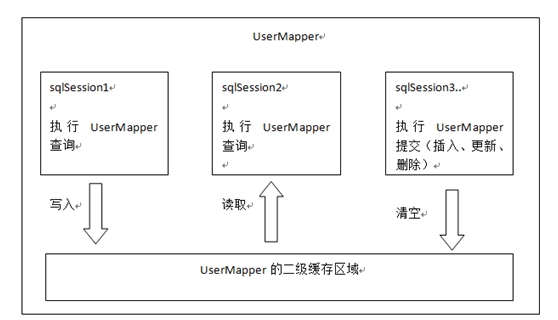


一级缓存区域是根据SqlSession为单位划分的。

每次查询会先去缓存中找，如果找不到，再去数据库查询，然后把结果写到缓存中。Mybatis的内部缓存使用一个HashMap，key为hashcode+statementId+sql语句。Value为查询出来的结果集映射成的java对象。

SqlSession执行insert、update、delete等操作commit后会清空该SQLSession缓存。

1. **二级缓存原理**



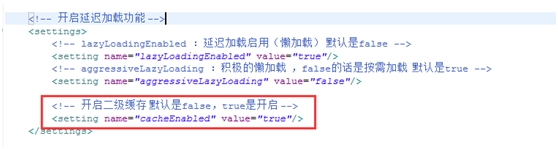
二级缓存是mapper级别的。Mybatis默认是没有开启二级缓存。

第一次调用mapper下的SQL去查询用户信息。查询到的信息会存到该mapper对应的二级缓存区域内。

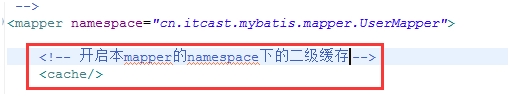
第二次调用相同namespace下的mapper映射文件中相同的SQL去查询用户信息。会去对应的二级缓存内取结果。

如果调用相同namespace下的mapper映射文件中的增删改SQL，并执行了commit操作。此时会清空该namespace下的二级缓存。

1. **开启二级缓存**
2. 在核心配置文件SqlMapConfig.xml中加入以下内容（开启二级缓存总开关）：cacheEnabled设置为 true



1. 在映射文件中，加入以下内容，开启二级缓存：



1. **实现序列化**

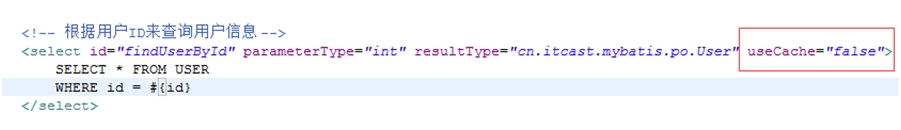
由于二级缓存的数据不一定都是存储到内存中，它的存储介质多种多样，所以需要给缓存的对象执行序列化。

如果该类存在父类，那么父类也要实现序列化。



1. **禁用二级缓存**

该statement中设置userCache=false可以禁用当前select语句的二级缓存，即每次查询都是去数据库中查询，默认情况下是true，即该statement使用二级缓存。



1. **刷新二级缓存**

