## Mybatis缓存机制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. 一级缓存：一级缓存mybatsi已经为我们自动开启，不用我们手动操作，而且我们是关闭不了的！！但是我们可以手动清除缓存。（SqlSession级别）   缓存不存在的条件：   |  |  | | --- | --- | | 一 | 并不是同一个Sqlsession对象 | | 二 | 在查询的过程中，两次的查询条件不同（其实这个不用我说大家也知道） | | 三 | 在两次之间进行了增，删，改，操作 | | 四 | 在两次之间自己进行了手动清空缓存 |   代码及其效果： |
| 1. 二级缓存：二级缓存需要我们手动开启。（全局级别）   二级缓存的作用：通过度一级缓存的了解。推挤缓存是基于同一个SqlSesion类的实例对象的。但是，有些时候在wenb工厂中将会执行查询操作的方法分装在某个Service方法中，当查询完一次后，Service方法结束，此时SqlSession类的实例对象就会关闭，一级缓存就会被清空。此时若再次调用用Service方法查询同一个信息，此时异界缓存就是空的，从而无法从缓存中获取信息！！         因此，我们可以使用二级缓存，二级缓存存在与Mapper实例中，当多个SqlSession类的实例对象加载相同的mapper文件，并执行其中国的SQL配置时，他们就共享一个Mapper缓存。当某个SqlSession类的实例对象执行了增，删，改，等改变数据的操作时，Mapper实例都会清空其二级缓存！  开启二级缓存：  开启二级缓存需要两个步骤：  第一步在mybatis的全局配置文件中配置Setting属性，设置名为cacheEnabled的属性值为true即可：   |  | | --- | | <settings>  <!--  (1):开启二级缓存,这个全局的配置二级缓存  默认是开启的，但是还是需要写上，防止版本的更新  -->  <setting name="cacheEnabled" value="true"/>  </settings> |   第二步：在具体需要二级缓存的mapeer映射文件中开启二级缓存，值需要在相应的映射文件中添加一个cache标签即可   |  | | --- | | (2):在相应的映射文件中开启二级缓存  <!-- 开启二级缓存 -->  <cache></cache> |   代码及其效果：    **二级缓存注意事项：**  **被缓存的实体类必须实现Serializable接口(使其可序列化)，并且使用第二个SqlSession对象之前不许关闭第一个Sqlsession**   1. **如果没有将实体列序列化，且没有关闭第一个SqlSession对象，则二级缓存无效** 2. **如果没有将实体类序列化，但关闭了第一个SqlSession，则会出现java.io.NotSerializableException: com.tom.entity.UserEntity这个异常，且缓存失效** 3. **如果将实体类序列化了，但是没有关闭第一个SqlSession，则缓存失效** |