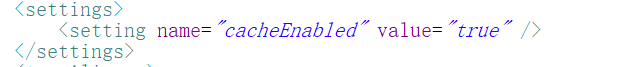
二级缓存操作步骤

第一步：开启二级缓存settings中配置

<setting name="cacheEnable" value="true"/>



第二部：使用二级缓存：mapper.xml使用<cache>

高级使用：制定淘汰算法（策略）、刷新时间、缓存对象个数、只读模式

<cache eviction="FIFO" flushInterval="6000" size="1024" readOnly="true"/>

eviction:淘汰策略

1.LRU-最近最少使用的：一处最长时间不被调用的对象。

2.FIFO-先进先出：按帝乡进入缓存的的顺序来移除他们。

3.SOFT-软应用：一处基于垃圾回收器状态和软引用规则对象。

4.WEAK-软引用：更积极地一处基于垃圾收集器状态和弱引用规则的对象

flushInterval：刷新的间隔 单位毫秒

size: 缓存对象的数量 默认1024

eadOnly：只读

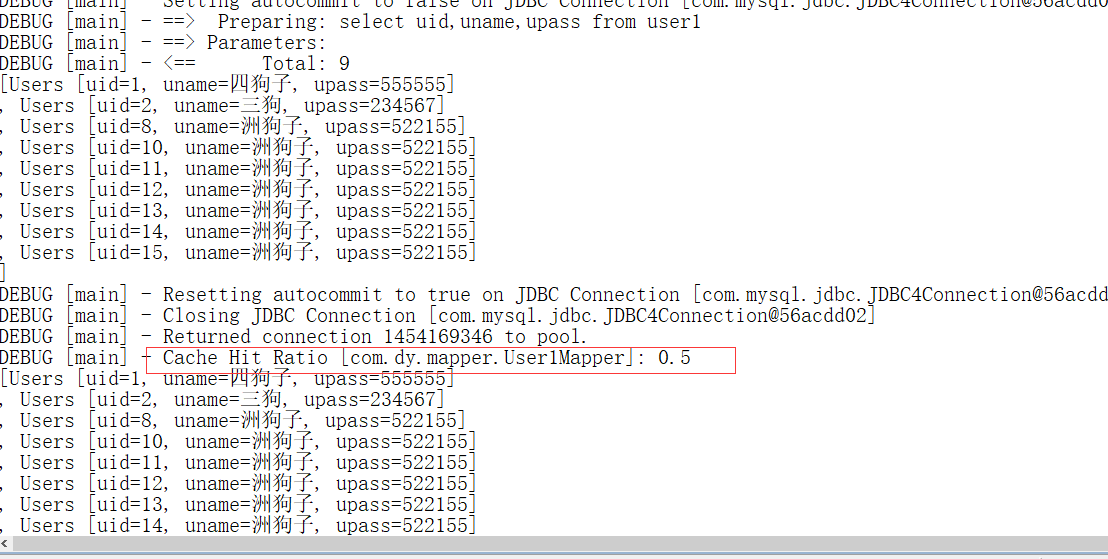
clipboard.png

第三步：将缓存对象的类序列化

实体类继承Serializable

clipboard.png

第四部：测试



红色框中是命中率

useCache配置

在statement中设置useCache=false可以禁用当前select语句的二级缓存，即每次查询都会发出sql去查询，默认情况是true，即该sql使用二级缓存。

<select id="findOrderListResultMap" resultMap="ordersUserMap" useCache="false">

总结：针对每次查询都需要最新的数据sql，要设置成useCache=false，禁用二级缓存。

刷新缓存（就是清空缓存）

刷新缓存就是清空缓存。在mapper的同一个namespace中，如果有其它insert、update、delete操作数据后需要刷新缓存，如果不执行刷新缓存会出现脏读。

设置statement配置中的flushCache="true"属性，默认情况下为true即刷新缓存，如果改成false则不会刷新。使用缓存时如果手动修改数据库表中的查询数据会出现脏读。如下：

<insert id="insertUser" parameterType="cn.itcast.mybatis.po.User" flushCache="true">

总结：一般下执行完commit操作都需要刷新缓存，flushCache=true表示刷新缓存，这样可以避免数据库脏读。

应用场景和局限性

应用场景

对于访问多的查询请求且用户对查询结果实时性要求不高，此时可采用mybatis二级缓存技术降低数据库访问量，提高访问速度，业务场景比如：耗时较高的统计分析sql、电话账单查询sql等。

实现方法如下：通过设置刷新间隔时间，由mybatis每隔一段时间自动清空缓存，根据数据变化频率设置缓存刷新间隔flushInterval，比如设置为30分钟、60分钟、24小时等，根据需求而定。

局限性

mybatis二级缓存对细粒度的数据级别的缓存实现不好，比如如下需求：对商品信息进行缓存，由于商品信息查询访问量大，但是要求用户每次都能查询最新的商品信息，此时如果使用mybatis的二级缓存就无法实现当一个商品变化时只刷新该商品的缓存信息而不刷新其它商品的信息，因为mybaits的二级缓存区域以mapper为单位划分，当一个商品信息变化会将所有商品信息的缓存数据全部清空。解决此类问题需要在业务层根据需求对数据有针对性缓存。