# 为什么需要二级缓存?

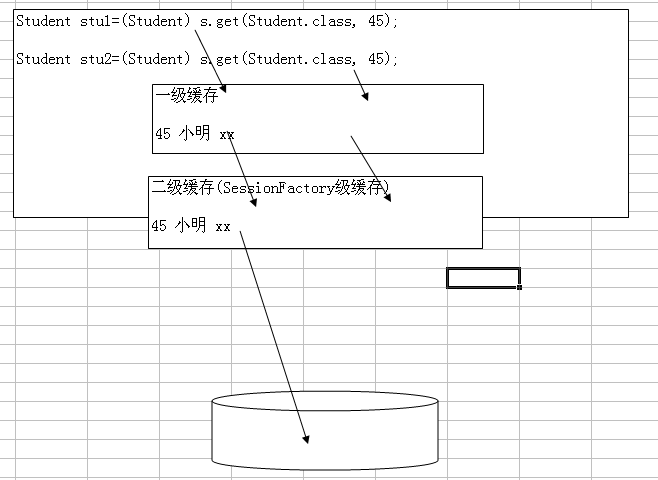
### 因为一级缓存有限(生命周期短),所以我们需要二级缓存(SessionFactory缓存)来弥补这个问题

## 1.1需要配置

## 1.2二级缓存是交给第三方去处理,常见的Hashtable , 1.3OSCache , EHCache



## 1.4二级缓存的原理



## 1.5二级缓存的对象可能放在内存，也可能放在磁盘.

# 快速入门案例

## 1.1使用OsCache来演示二级缓存的使用.

## 1.2开始配置，记得需要引入jar包

### 2.1配置在config.xml中与下面这个是或者关系

对配置说明:

<property name="hbm2ddl.auto">update</property>

<!-- 启动二级缓存 -->

<property name="cache.use\_second\_level\_cache">true</property>

<!-- 指定使用哪种二级缓存 这里要引入jar包-->

<property name="cache.provider\_class">org.hibernate.cache.OSCacheProvider</property>

<mapping resource="com/hsp/domain/Department.hbm.xml" />

<mapping resource="com/hsp/domain/Student.hbm.xml" />

<!-- 指定哪个domain启用二级缓存

特别说明二级缓存策略:

1. read-only

2. read-write

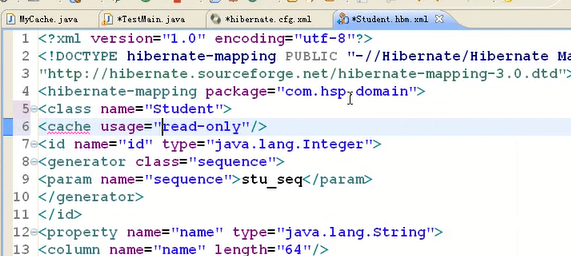
3. nonstrict-read-write

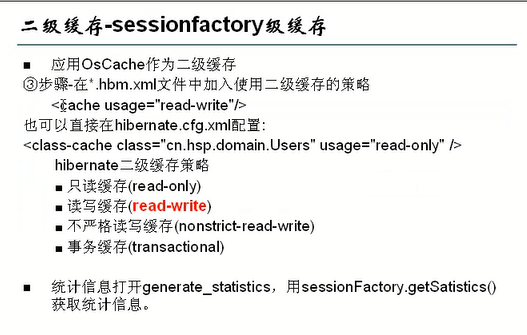
4. transcational

-->

<class-cache class="com.hsp.domain.Student" usage="read-write"/>

### 2.2配置在domain的 hbm.xml中





## 1. 3可以文件放在 src目录下，这样你可以指定放入二级缓存的对象capacity 大小. 默认1000



# 进行测试

## 1.1代码

**public** **class** HuanCun {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Session s=**null**;

Transaction tx=**null**;

**try** {

//我们使用基础模板来讲解.

s=HibernateUtil.*openSession*();

tx=s.beginTransaction();

//查询45号学生

StudentMany stu1=(StudentMany) s.get(StudentMany.**class**, 2);//45->一级缓存和二级缓存中放入，也就错过（没有找到），和放入（二级中）

System.*out*.println(stu1.getName());

tx.commit();

} **catch** (Exception e) {

e.printStackTrace();

**if**(tx!=**null**){

tx.rollback();}

}**finally**{

**if**(s!=**null** && s.isOpen()){

s.close();}}

System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**try** {

//我们使用基础模板来讲解.

s=HibernateUtil.*openSession*();

tx=s.beginTransaction();

这个就是在二级缓存中找，命中

StudentMany stu1=(StudentMany) s.get(StudentMany.**class**, 2);

System.*out*.println(stu1.getName());

下面这个session没有关闭，所以一级缓存张找，没有任何命中错过

StudentMany stu1=(StudentMany) s.get(StudentMany.**class**, 2);

System.*out*.println(stu1.getName());

下面这个就会到数据库中找，错过和放入

StudentMany stu3=(StudentMany) s.get(StudentMany.**class**, 3);

System.*out*.println(stu3.getName());

tx.commit();

} **catch** (Exception e) {

e.printStackTrace();

**if**(tx!=**null**){

tx.rollback();

}

}**finally**{

**if**(s!=**null** && s.isOpen()){

s.close();

}

}

//完成一个统计，统计的信息在Sessfactory

//SessionFactory对象.

Statistics statistics= HibernateUtil.*getSessionFactory*().getStatistics();

System.*out*.println(statistics);

System.*out*.println("放入"+statistics.getSecondLevelCachePutCount());

System.*out*.println("命中"+statistics.getSecondLevelCacheHitCount());

System.*out*.println("错过"+statistics.getSecondLevelCacheMissCount());

}

## 2.进行一个统计 //完成一个统计，统计的信息在Sessfactory

### 这个是需要在config中进性配置的



//完成一个统计，统计的信息在Sessfactory

//SessionFactory对象.

Statistics statistics= HibernateUtil.*getSessionFactory*().getStatistics();

System.*out*.println(statistics);

System.*out*.println("放入"+statistics.getSecondLevelCachePutCount());

System.*out*.println("命中"+statistics.getSecondLevelCacheHitCount());

System.*out*.println("错过"+statistics.getSecondLevelCacheMissCount());

## 3. 在配置了二级缓存后，请大家要注意可以通过 Statistics,查看你的配置命中率高不高，命中率越高越好