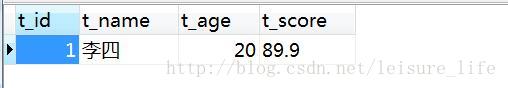
# Hibernate缓存机制之快照

（1）、快照是数据的副本   
（2）、快照属于一级缓存   
（3）、快照是在堆内存中的   
（4）、快照的作用：保证数据一致性   
当执行`session.getTransaction().commit()时，**[hibernate](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/leisure_life/article/details/_blank)**同时会清理session的一级缓存（flush），也就是将堆内存中的数据与快照中的数据进行对比，如果不一致，则会执行同步（update）操作，若相同，则不执行update。

## 举个栗子

**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/leisure_life/article/details/_blank)**中有一条数据：   
   
**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/leisure_life/article/details/_blank)**代码：

public void testGet(){

//获取session 对象

Session session = HbnUtils.getSession();

//开启事务

session.beginTransaction();

try {

//执行get操作

Student student = session.get(Student.class, 1);

student.setName("王五");

System.out.println(student);

//提交事务

session.getTransaction().commit();

} catch (Exception e) {

//回滚事务

session.getTransaction().rollback();

e.printStackTrace();

}

}

## 日志信息

select

student0\_.t\_id as t\_id1\_0\_0\_,

student0\_.t\_name as t\_name2\_0\_0\_,

student0\_.t\_age as t\_age3\_0\_0\_,

student0\_.t\_score as t\_score4\_0\_0\_

from

t\_student student0\_

where

student0\_.t\_id=?

Student [id=1, name=王五, age=20, score=89.9]

Hibernate:

update

t\_student

set

t\_name=?,

t\_age=?,

t\_score=?

where

t\_id=?

## 分析

我们都知道，执行了get方法之后，DB中的数据就加载到session缓存中来了，而执行student.setName("王五")本来应该只是改变了session缓存（堆内存）中的数据，为什么数据库的数据也改变了。   
再来解剖一下Student student = session.get(Student.class, 1);`   
1）、将数据从DB中取出来

2）、将数据转变成对象，并存入堆内存中

3）、将对象的id放入session缓存map的key中，将对象的引用放入session缓存map的 value中，这就纳入session管理了   
4）、将对象的详情放入到“快照”中   
当执行了`session.getTransaction().commit();时，Hibernate为了保证数据的一致性，Hibernate会清理session的一级缓存（flush），也就是将堆内存中的数据（已经纳入session管理的数据）与快照中的数据进行对比，如果不一致，则会执行同步（update）操作，若相同，则不执行update。

由于在commit()前，我们执行了student.setName("王五");导致堆内存中是数据与快照中的数据不一致，所以它执行了update，以便保证数据的一致性。

## 为什么需要快照

通过上面的分析知道了快照的根本作用是保证数据一致性，保证数据一致的另一种做法是，commit之前把堆内存中的数据直接与数据库中的对应记录进行对比，显而易见这样的效率是灰常低下的，而采用快照技术，因为快照是一定和数据库中记录一致的，快照也在堆内存中，所以速度不是一般的快。