

Hibernate02

一. 什么是缓存?

简单的说就是一块内存空间,将数据源（数据库或者文件）中的数据存放到缓存中.再次获取的时候,直接从缓存中获取.

但是有很多问题:

1. 有一定方法将数据放入到缓存中
2. 有一些方法从缓存中提取数据
3. 如果缓存中的数据发生变化, 需要把数据同步到数据库中
[如果有很多数据, 有的变化了, 有的没有变化.] 检查缓存中数据是否和数据库中数据一致
4. 把数据库中数据同步到缓存中
5. 内存的空间有限, 也不能无限的放入, 就需要定期清除. 有一个词: hits 命中率. 命中率越低的对象应该从缓存中移除.

二. Hibernate 一级缓存

2.1 概念

- 一级缓存也称为 session 级别的缓存.
- 一级缓存的生命周期和 session 生命周期一致
- 自带的不可卸载的

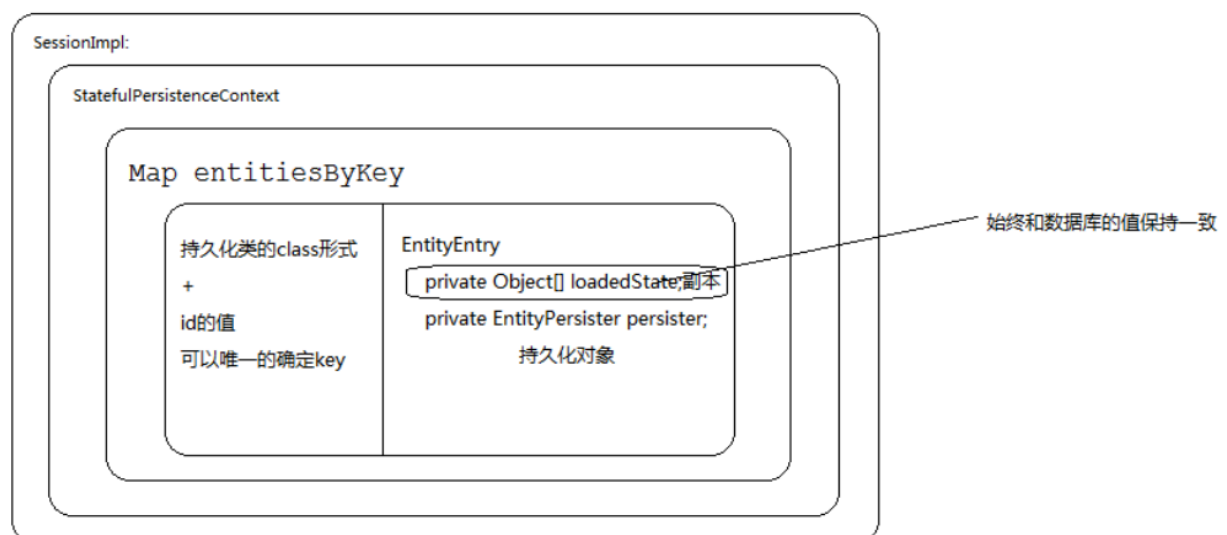
2.2 位置

- SessionImpl / StatefulPersistenceContext / private Map<EntityKey, Object> entitiesByKey;
- 这个 Map 就是一级缓存
- 如果说一个对象是持久化状态对象，这个对象在一级缓存中。

2.3 操作 Hibernate_02_1_SessionCache

- get . save . update . evict . clear . close

2.4 一级缓存的内存结构



[查看文档](#) -- [一级缓存查看源码](#)

三. Session

3.1 创建 session 的方式

3.1.1 openSession

- 重新创建了一个新的 session，就打开了一个新的连接
- 但是，在有些场景必须关注session的创建，比如说在银行转账操作的时候，两个账户转账必须在同一事务中，也就必须在同一个session中；而 openSession 做不到

3.1.2 getCurrentSession

- 先检查当前线程中是否有session
- 如果当前线程中有session,则把session提取出来，直接使用
- 如果当前线程中没有session,则采用openSession方法创建session
- 把新创建的session放入到 ThreadLocal 中
- 当再次得到session的时候就是从当前线程中获取了

3.2 getCurrentSession 的使用

- 1.需要在 hibernate.cfg.xml 中配置
 - `<property name="current_session_context_class">thread</property>`
- 2.如果使用 getCurrentSession 的话，CRUD 的操作全部都要在事务当中进行
- 3.如果使用 getCurrentSession 的话,当事务提交的时候，session自动关闭
 - 这种做法相当于把session和事务绑定在一起了

四. 一对多单向

Classes 能够找到 Student,但是Student 找不到 Classes

注意：这里在实体类中是在one的一方维护了一个集合，但是在数据库中是在many的一方维护了一个指向one主键的外键。

```
<one-to-many class="com.lanou.dao.ontomany.Student"/>
```

4.1 操作 Hibernate_02_2_OneToMany

4.1.1 一般操作

1. 保存班级

2. 保存学生
3. 保存班级, 学生

4.1.2 级联操作和关系操作

- cascade指的是级联操作, 操作的是一般属性[除标识符属性和关联对象属性], 指的是对象与对象的操作
 - save-update : 在保存 classes 对象的时候, 针对student进行保存或者更新的操作, 在更新 classes 对象的时候, 针对student进行保存或者更新的操作
 - delete : 级联删除
 - all : 级联保存、更新、删除
- inverse指的是关系操作, 针对的就是外键
 - true : classes 不维护 classes 和 student 之间的关系
 - false : classes 维护 classes 和 student 之间的关系 [默认]

级联操作

1. 保存班级级联保存学生
2. 保存班级级联更新学生
3. 更新班级级联保存学生
4. 更新班级级联更新学生
5. 删除班级级联删除学生
6. 在班级有级联 save-update 的情况下, 从关联得到学生并删除?

关系操作 -- inverse

1. 已经存在一个班级, 新建一个学生, 把学生加入到该班级
 2. 已经存在一个学生, 新建一个班级, 把学生加入到该班级
 3. 已经存在一个班级, 已经存在一个学生, 把学生加入到该班级
 4. 已经存在一个学生, 把一个学生从另一个班级cid=2转入该班级cid=1
 5. 解除班级和学生之间的关系
 6. 解除班级和学生之间的关系, 再建立该班级和一些学生的关系
- * 级联和关系混合
1. 在删除班级的时候, 解除班级和学生之间的关系

五. 一对多双向

当多的一方维护关系时, 不会发出更新关系的 update 语句, 而一的一方维护关系时, 需要发出维护关系的 update 语句. 一般情况下, 多的一方维护关系效率更高. [特殊情况: 一个班级解除所有学生关系. 这就不能从学生入手]

没有 inverse 属性, 因为操作了 student 本身了, 没有存在维护不维护外键一说


```
<many-to-one name="clazz" column="cid" class="com.lanou.domain.Clazz"
cascade="save-update"/>
```

5.1 操作 Hibernate_02_3_OneToManyBoth

六.多对多

一对多描述了对象和集合关系, 多对一描述了对象和对象关系, 多对多描述了集合和集合的关系.

维护关系: 多对多建立关系相当于在第三张表中插入一条数据. 解除关系就是在第三张表中删除一条数据.

多对多从谁那维护都一样, 看需求

6.1 操作 Hibernate_02_4_ManyToMany

七.一对一

一对多双向中 student 中的 cid 外键是唯一.

unique="true": 外键只能在这张表里出现一次

7.1 操作 Hibernate_02_5_OneToOne

