韩顺平Hibernate讲解

Hibernate：

**1、hibernate是什么?**

1）hibernate是一个框架(framework)

2）hibernate是一个orm框架（对象关系映射框架）【orm:object relation mapping】

3）Hibernate处于我们项目的持久层位置（正因为如此，所以又把Hibernate称为持 久层框架）。

4）对象持久化：就是把对象的信息保存到数据库或文件中。

5）Hibernate实际上就是对JDBC进行了轻量级的封装。

6）hibernate 的基础还是我们java 反射机制

7）Hibernate的核心包括三个：HibernateAPI、hibernate.cfg.xml核心配置文件、 对象关系映射文件。

**2、为什么需要hibernate：**

1）我们使用的数据库是关系数据库，也就是说是关系模型，而Java却是一个个的对 象，也就是对象模型。当Java访问数据库的时候，是由对象模型到关系模型，这 之间有一层阻抗，hibernate的作用就是消除这一层阻抗。

**3、使用Hibernate的好处：**

1. 引入Hibernate可以使工作人员角色细化，让程序员更关心业务流程，让数据库人 员关注数据库相关的各种操作。
2. 分层清晰，耦合性小
3. 通用性强：可以更轻松的从一个数据库平台转移到别的平台
4. 对象化：把关系数据库变成了Java的对象，更加方便操作。
5. 性能保证：Hibernate可以按不同的数据库、处理不同的操作、使用最优的SQL 语句，不用我们去想。对于分等算法，在Hibernate中会显得更简答、可靠。
6. 增加了程序的鲁棒性。

**4、orm框架除了hiberante 之外，还有一些： apache ojb / toplink / ibatis / ejb cmp**

快如入门案例:【参考hibernate1-1】

**1、开发流程：**

1）创建一个项目

2）画出一个简单项目框架示意图

3）引入hibernate开发包

4）开发hibernate 有三种开发顺序，我们使用先开发DB（数据表），然后在开发 mapping文件以及domain类。

5）创建employee2 表

|  |
| --- |
| create table employee2(  id number primary key,  name varchar2(64) not null,  email varchar2(64) not null,  hiredate date not null  ) |

1. 开发domain对象和mapping文件（对象关系映射文件）
2. mapping文件（对象关系映射文件）：作用是用于指定domain对象和表的映射关 系。该文件名称建议规范：domain对象**.**hbm**.**xml。一般放在和domain对象同 一个包下。

在Employee.hbm.xml文件中有下面的一段配置，所以需要在数据表中创建一个 sequence。

|  |
| --- |
| <generator class=*"sequence"*>  <param name=*""*></param>  </generator> |

在employee表中创建sequence（序列）：用于主键的自增长

|  |
| --- |
| create sequence employee\_seq  start with 1  increment by 1  minvalue 1  nomaxvalue  nocycle  nocache |

1. 配置hibernate.cfg.xml文件，该文件用于配置连接的数据库的类型、driver、用 户名、密码、URL，同时还可以管理对象映射文件。我们可以在 F:\Java插件 \Hibernate\hibernate-release-4.2.21.Final\project\etc 目录下的 hibernate.properties 文件中查看关于连接各种数据库的各种配置。
2. 因为SessionFactory是重量级的，及其消耗内存，所以一定保证只有一个 SessionFactory。一个数据库对应一个SessionFactory对象。为此，我们自己定 义了一个MySessionFactory类，用来实例SessionFactory对象。

**2、如何回滚事务：【参考TestMain2.java】**

**3、遇到的问题：**

1. 在开发的时候，在修改数据的时候遇到了下面的一个错误

domain.Employee\_$$\_javassist\_0 cannot be cast to javassist.util.proxy.Proxy

暂时还不知道是什么原因，但是挡在Employee.hbm.xml文件从class节点配置上 lazy=false属性，就可以正常运行。

|  |
| --- |
|  |

POJO：简单Java类

1、pojo类是和一张表对应

2、一般我们放在 com.xxx.domain下

3、pojo 需要一个主键属性(用于标示一个pojo对象)

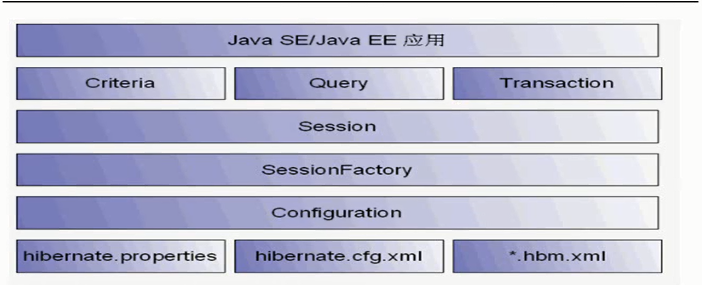
4、除了主键属性外，它应当还有其属性，属性的访问权限是private

6、提供 set /get 方法

7、它应当有一个无参的构造方法(hibernate 反射)

8、pojo类其实就是javabean/ 有些老程序员 叫他 data对象

hibernate的核心类和接口：（参考hibernate1-2）



**1、Configuration 类**

1）Configuration类的作用

①读取hibernate.cfg.xml

②管理对象关系映射文件 <mapping resource=””>

③加载hibernate 的驱动，url ，用户..

④管理hibernate配置信息

1. 我们一般都是将hibernate.cfg.xml文件放在src目录同级下，但是也有很多人专 门将各种配置文件放置在诸如configuration这样单独的包下，如果是这样，那怎 么解决？

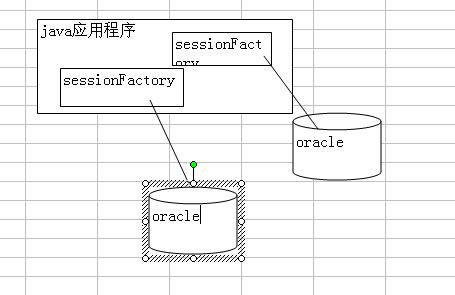
【参考 hibernate1-2】

**2、SessionFactory：会话工厂**

1）可以缓存sql语句和数据(称为session级缓存，以及缓存)!!

2）是一个重量级的类（很消耗内存），一般用单例模式保证一个应用中只需要一个 SessionFactroy

3）如果某个应用访问多个数据库，则要创建多个会话工厂实例，一般是一个数据库一 个会话工厂实例。



1. 通过SessionFactory接口可以获得Session（会话）实例。而获取session有两个 方法：openSession()、getCurrentSession()
2. openSession()与getCurrentSession()

1、openSession()方法：每次调用openSession()方法会新创建一个Session对 象。

2、getCurrentSession()方法：获取和当前线程绑定的Session，换言之，在同一 个线程中，我们获取的session是同一个，这样可以利于事务的控制。如果希 望使用getCurrentSession()方法则需要在hibernate.cfg.xml文件中配置。

3、采用getCurrentSession()创建的session在commit或rollback时会自动关 闭，但是我们还是建议自己再手动关闭一下；而采用openSession()创建的 session必须手动关闭。

4、在SessionFactory启动的时候，hibernate会根据配置创建相应的 CurrentSessionContext，在getCurrentSession()被调用的时候，实际被执行的 方法是CurrentSessionContext().currentSession()。在currentSession()执行时， 如果当前Session为空，currentSession会调用SessionFactory的openSession。

【也就是说如果没有session，调用getCurrentSession()会自动调用openSession()方法，创建一个session。】

5、选择的原则：

①如果需要在同一线程中，保证使用同一个Session则，使用getCurrentSession()

②如果在一个线程中，需要使用不同的Session,则使用opentSession()

6、可以在 cmd 中 使用“netstat -an”查看端口占用情况。

**3、Session（会话）接口：**

1）Session一个实例代表与数据库的一次操作(当然一次操作可以是crud组合)

2）Session实例通过SessionFactory获取，用完需要关闭。

3）Session是线程不同步的(不安全),因此要保证在同一线程中使用,可以用 getCurrentSessiong()。

4）Session可以看做是持久化管理器,它是与持久化操作相关的接口

5）Session一般以对象的形式来操作：

①保存一个对象（记录）：save()方法

②删除一个对象（记录）：delete()方法

③查询一个对象（记录）：get()/load()方法

④修改一个对象（记录）：update()方法

1. get()方法和load()方法的区别：【参考TestMain2**.**java】

①get()方法直接返回实体类。如果查询不到数据则返回null；load()方法会返回 一个实体类代理对象（当前这个对象可以自动转换为实体对象），但当代理对 象被调用时，如果发现数据不存在，就会抛**ObjectNotFoundException异常**

②load()方法先到缓存（session缓存/二级缓存）中去查，如果没有则返回一个 代理对象（不马上到DB中去查），等后面使用这个代理对象操作的时候，才 到DB中查询，这就是我们常说的“load在默认情况下支持延迟加载”。

③get先到缓存（session缓存/二级缓存）中去查，如果没有就到DB中去查（即 马上发出sql）。总之，如果你确定DB中有这个对象就用load()方法，不确 定就用get()方法。

④使用get 去查询数据，(先到一级/二级)会立即向db发出查询请求(select ...), 如果你使用的是 load查询数据，(先到一级、二级))即使查询到对象，返回的 是一个代理对象,如果后面没有使用查询结果，它不会真的向数据库发select , 当程序员使用查询结果的时候才真的发出select ,这个现象我们称为懒加载 (lazy)

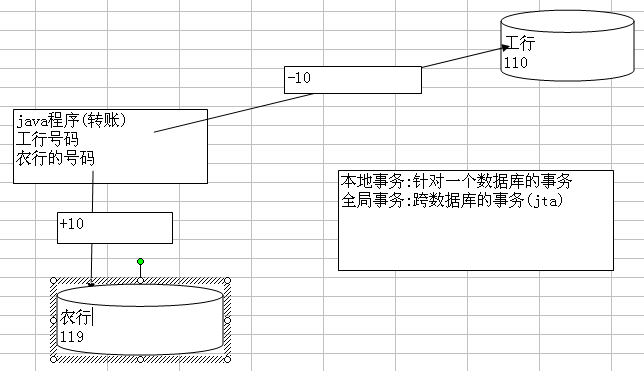
⑤【注：一级缓存（Session缓存）是必须使用的，二级可用可不用，需要我们 手动配置】

1. **Query接口：**
2. 之前的load()或者get()方法都是根据主键才能查询数据表中的数据，而如果要根 据其它字段从查询，那么load()、get()方法就不适用了，此时就需要使用Query() 接口中的方法了。
3. 【案例：通过name查询数据】【参考TestMain3**.**java】

3、

全局事务和本地事务

**1、图解**



**2、配置**

1）本地事务：

|  |
| --- |
| <property name=*"current\_session\_context\_class"*>thread</property> |

1. 全局事务：

|  |
| --- |
| <property name=*"current\_session\_context\_class"*>jta</property> |

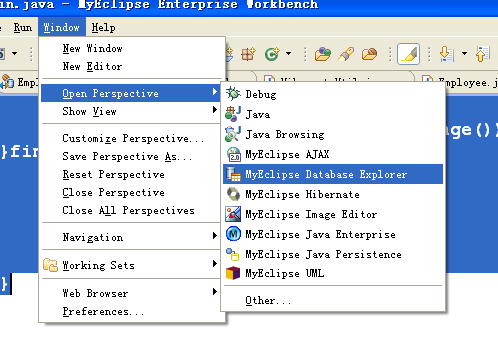
利用MyEclipse快速开发hibernate项目【参考hibernate1-3】

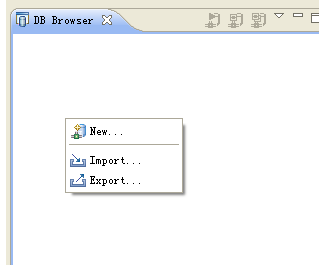
1、手动配置：db(table )-> 手写domain对象->对象关系映射文件

2、MyEclipse生成：

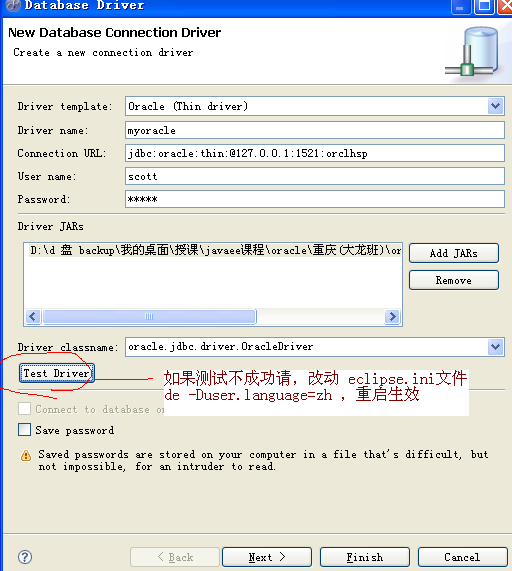
1）创建web项目

2）通过myeclipse 提供 数据库浏览器连接到我们的oracle数据库(多人开发)



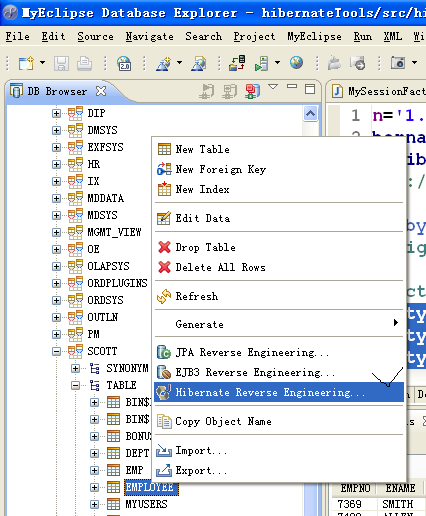


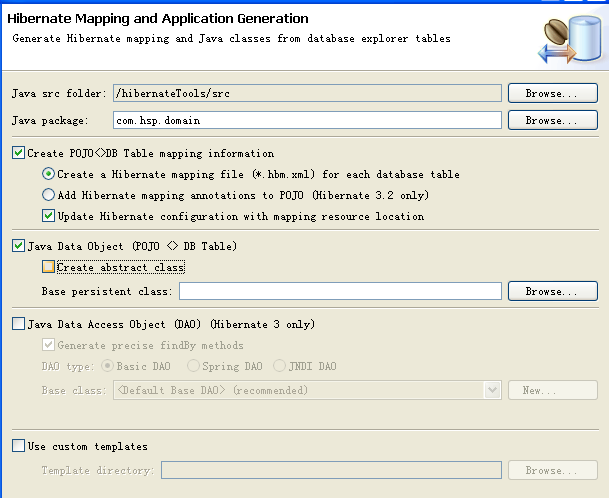
3）这里请大家小心，如果我们测试



4）引入hibernate开发包.,同时自动创建hibernate.cfg.xml文件,如果希望把hibernate 开发包升级，我们可以重新引入包.

5）下面我们使用myeclipse提供的逆向工程，自动的创建domain类和对象关系映射文件.





1. **问题：**

1）当测试的时候，会报错：Error applying BeanValidation relational constraints

原因是因为：javax.persistence.validation.mode默认情况下是auto的，就是说如果不设置的话它是会自动去你的classpath下面找一个bean-validation\*\*包，但是找不到，所以beanvalitionFactory错误。

所以需要在hibernate.cfg.xml文件中加上下面的配置：

|  |
| --- |
| <property name=*"javax.persistence.validation.mode"*>none</property> |

2）当利用hibernate的反向工程生成 \*.hbm.xml 文件的时候，会多出下面一个空格。

|  |
| --- |
|  |

所以需要将那个空额删除！

HQL：【参考hibernate1-4/TestMain.java】

1. HQL是面向对象的语言，与SQL不同，HQL中的对象名的区分大小写的；HQL中查的 是对象而不是表，并且支持多态，HQL主要是通过Query来操作。
2. uniqueResult()方法：当查询的数据只有一条的时候，可以使用uniqueResult()方法【效 率更高】，但是如果是多条数据，使用该方法就会抛出异常。
3. 聚集（聚合）函数的使用【参考TestMain2**.**java】

4、分页【参考TestMain3**.**java】

5、参数绑定【参考TestMain4.java】

1）使用参数绑定的好处：

①可读性好

②性能提高

③防止sql注入

1. 参数绑定有两种形式：

①冒号形式：

|  |
| --- |
| Query query = session.createQuery("from Student where sdept=:param1 and sage>:param2");  query.setString("param1", "计算机系");  query.setInteger("param2", 20);  List<Student> list = query.list(); |

②问号形式：

|  |
| --- |
| Query query2 = session.createQuery("from Student where sdept=? and sage>?");  query2.setString(0, "外语系");  query2.setInteger(1, 20);  List<Student> list2 = query2.list(); |

1. 在映射文件中配置HQL语句

1）把HQL语句配置到 对象关系映射文件中

|  |
| --- |
| <query name=”myquerytest”>  <![CDATA[select sname from Student where sage>22]]>  </query> |

1. 在程序中获取并执行

|  |
| --- |
| List list = session.createQuery(“myquerytest”).list(); |

1. 使用子查询【参考TestMain5**.**java】

Hibernate对象的三种关系：

1. one-to-one：
2. one-to-many：
3. many-to-many：

hibernate开发的三种方式中的：

1、这里我们说明如果要自动的创建出对应的数据库，需要做配置(hibernate.cfg.xml).

**<property name="hbm2ddl.auto">create</property>**

这里有四个配置值: create , update , create-drop, validate

1）create : 当我们的应用程序加载hibernate.cfg.xml [ new Configuration().config()]就会根据映射文件，创建出数据表, 每次都会重新创建【先 删除原有的表，再创建新表】

2）update： 如果数据库中没有该表，则创建，如果有表，则看有没有变化，如果有 变化，则更新（如新增一列、修改字段的取值范围、添加数据等）。

3）create-drop：首先创建对应的表，然后在显式（手动）关闭 sessionFactory（不 是session）时，在删除之前创建的表。

4）validate：相当于每次插入数据之前都会验证数据库中的表结构和hbm文件的结构 是否一致

2、使用建议：

在开发测试中，我们配置哪个都可以测试，但是如果项目发布后，最好自己配置一次， 让对应的数据库生成，完后取消配置，

domain对象的细节：

1、需要一个无参的构造函数(用于hibernate反射该对象)

2、应当有一个无业务逻辑的主键属性.

3、给每个属性提供 getter、 setter方法.

4、在domian对象中的属性，只有配置到了对象映射文件后，才会被hiberante管理.

5、属性一般是private范围

hibernate对象的三种状态【参考hibernate1-4】

1. 三种状态：【参考TestMain6**.**java】

1）瞬时状态（transient）：数据库中没有数据与之对应，超过作用域会被JVM垃圾 回收器回收，一般是new出来且与session没有关联的对象。

2）持久状态（persistent）：数据库中有数据与之对应，与当前session有关联，并 且相关联的session没有关闭，事务没有提交；持久对象状态发生改变，在事务提 交时会影响到数据库（hibernate能检测到）  
 3）脱管/游离状态（detached）：数据库中有数据与之对应，但当前没有session与 之关联；脱管对象状态发生改变，hibernate不能检测到。

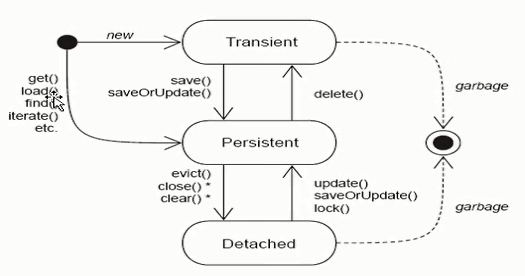
2、如何判断一个对象处于怎样的状态？

1）主要的依据是：

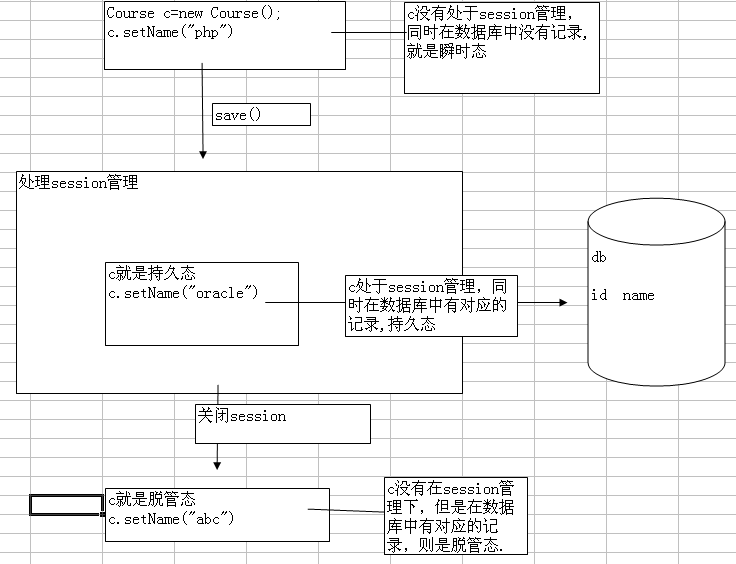
①看该对象是否处于session管理之下。

②看在数据库中有没有对应的记录

1. 对象状态转换图：



4、对象状态详解图：



1. 注意点：
2. delete()后数据被删除、且不在session管理之下，所以此时是瞬时态。
3. commit()提交之后，就处于脱管态。

关系映射：【参考hibernate1-5、hibernate1-6】

1、many-to-one 【参考hibernate1-5/TestMain**.**java】

2、one-to-many 【参考hibernate1-6/TestMain.java】

Department表中：

|  |
| --- |
| **private** Set<Student2> stus; |

Department.hbm.xml配置文件中：

|  |
| --- |
| <!-- 配置one-to-many关系 -->  <set name=*"stus"*>  <!-- 指定外键 -->  <key column=*"dept\_id"*></key>  <one-to-many class=*"Student2"*/>  </set> |

1. one-to-one

1）基于主键的one-to-one：【参考hibernate1-7】

2）基于外键的one-to-one：【参考hibernate1-8】

4、many-to-many：【参考hibernate1-9】

1）在操作和性能上都不太理想，所以多对多的映射使用较少，实际上使用时最好最换 成一对多的对象模型；hibernate会为我们创建中间关联表，转换成两个一对多（多 对一）。

5、级联操作（cascade）：

1）cascade用来说明当对主对象进行某种操作时是否对其关联的从对象也做相关的操 作。

2）常用的cascade：

none、all、save-update、delete、lock、refresh、evict、replicate、persist、 merge、delete-orphan（one-to-many）。

1. 一般在many-to-one、many-to-many不设置级联，在one-to-one、 one-to-many中设置级联。
2. 使用范围：

①在集合属性和普通属性中都可以使用cascade

②一般在one-to-one、one-to-many中配置cascade。

1. 案例：当删除某个部门的时候，该部门中的所有学生都会被删除【参考hibernate1-10】

|  |
| --- |
| <!-- 配置one-to-many关系 -->  <!-- cascade="delete": 当删除某部门(主对象)的时候,则级联删除该部门的所有学生 -->  <set name=*"stus"* cascade=*"delete"*>  <!-- 指定外键 -->  <key column=*"dept\_id"*></key>  <one-to-many class=*"Student3"*/>  </set> |

3、懒加载（lazy）

1）概述：

①懒加载是一种独特而又强大的数据获取方法，是指程序推迟访问数据库，这样做 可以保证有时候不必访问数据库，因为访问一次数据库是比较耗时的。

②当我们查询一个对象的时候，在默认的情况下，返回的是对象的普通属性，当用 户去使用对象属性时，才会向数据库发出再一次的查询，这种现象称之为lazy 现象！

1. 解决方法：

①显式初始化：Hibernate.initized(代理对象);

②修改对象关系文件 lazy=false

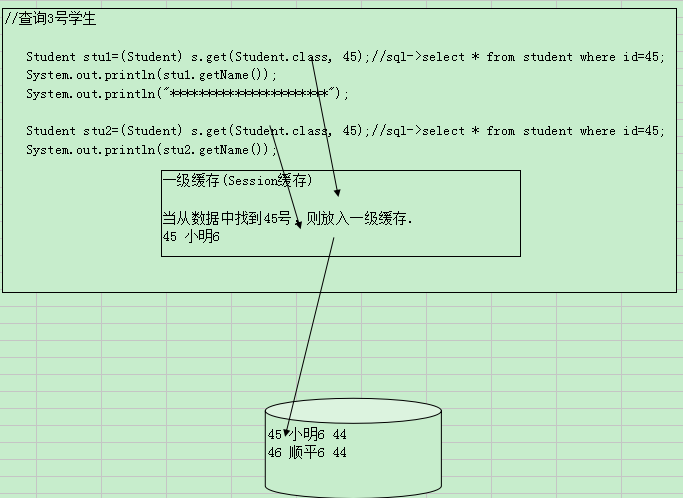
③通过过滤器 openSessionInView

缓存：

1、缓存的作用主要是用来提高性能，可以简单的理解成一个Map；使用缓存涉及到三个操 作：把数据放入缓存、从缓存中获取数据、删除缓存中的无效数据。

2、一级缓存（Session级共享）：

1）图解：

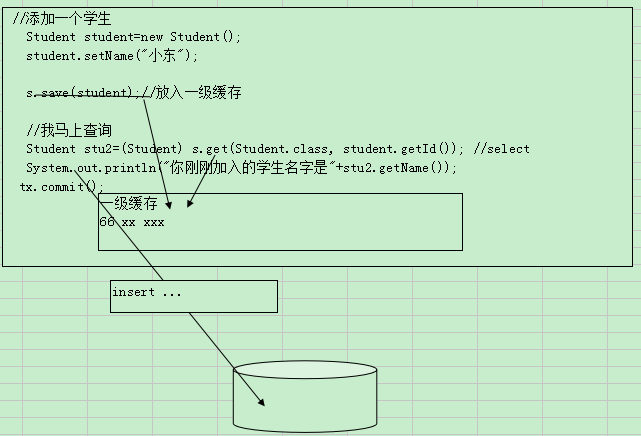


1. 说明：当我们去查询对象的时候，首先到一级缓存去取数据，如果有，则不到数据 库中取，如果没有则到数据库中取，同时将对象放入一级缓存中。
2. 细节：

A）什么操作会向一级缓存放入数据？

①save、update、saveOrUpdate、load、get、list、iterator、lock

②save举例：当执行s.save()的时候，只是会向数据库查询id，而后将Student 对象放入一级缓存中，而还没有放入数据库中。当执行到tx.commit()的时候， 才会将一级缓存的数据放入数据库中。



B）什么操作会从以及缓存中取出数据？

①get、load【注：list只会向以及缓存放入数据，不会从一级缓存取出数据， 而是直接查询数据库】

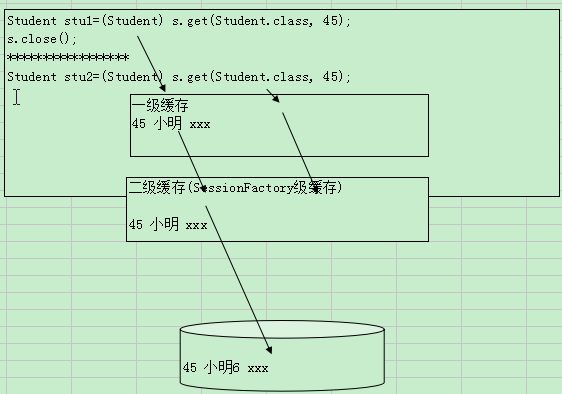
C）一级缓存不需要配置就可以使用；一级缓存本身没有保护机制；一级缓存不能 控制缓存的数量，所以要注意大批量操作数据时可能造成内存溢出；可以用 evict、clear方法清除缓存中的内容。

D）session级缓存的生命周期：当session关闭后，就自动销毁。

1. 二级缓存（SessionFactory级共享）：
2. 二级缓存需要配置；hibernate将二级缓存交给了第三方开发。常见的有 Hashtable、OSCache、EHCache。
3. 二级缓存的对象可能放在内存，也可能放在磁盘

3）二级缓存的原理：

A）图解：



B）说明：一级缓存没有，就到二级缓存中找，二级缓存没有，就到数据库中找； 数据库中找到之后，就将对象保存在二级缓存、一级缓存中。

C）疑问：既然二级缓存保存对象之后同时也会将对象保存在一级缓存中，那为什 么还需要二级缓存？

答：①一级缓存没有保护机制，不能控制缓存的数量等...

②以及缓存的生命周期短，是session级别的，session关闭，一级 缓存中对象就会被清除；而二级缓存是SessionFactory级别的，生 命周期更长。

1. 案例：使用EhCache来演示二级缓存的使用。【参考hibernate1-11】

|  |
| --- |
| <!-- 配置二级缓存 -->  <!-- 表示启用二级缓存 -->  <property name=*"cache.use\_second\_level\_cache"*>true</property>  <!-- 指定使用哪种二级缓存 -->  <property name=*"hibernate.cache.region.factory\_class"*>org.hibernate.cache.ehcache.EhCacheRegionFactory</property>  <!-- 指定哪个domain对象启用二级缓存 -->  <class-cache usage=*"read-write"* class=*"domain.Student3"*/> |

注意：①配置的先手顺序；②上面红色标记的要用这个（这是针对hibernate4.x的，若是hibernate3.x又是不一样的配置）

主键增长策略：

1. 什么是对象标识符：（OID：Object ID）

Hibernate中的持久化对象对应数据库中的一张数据表，因此区分不同的持久化对象， 在hibernate中是通过OID来完成的，从表的角度看，OID对应表的主键。从类的角 度看，OID对应类的主键属性。

2、对象表示符生成方法（主键增长策略）：

1）increment：由hibernate自动以递增的方式生成标识符，每次增量为1

优点：不依赖底层数据库系统，适用于所有的数据库系统。

缺点：适用于单进程环境下，在多进程环境下可能生成相同主键值。而且OID必 须为数值类型，long、short、int。

1. identity：由底层数据库生成标识符。

前提条件：数据库支持自动增长字段类型，每次增长1，比如（SQL Server，MySQL） 【Oracle不支持】，而却OID必须为数值类型，比如long，int，short

3）sequence：

4）native：native生成器能根据底层数据库系统的类型，自动选择合适的标识符生成 器，因此非常适用于跨数据库平台开发，它会由hibernate根据数据库适配器中 的定义，自动采用identity、hilo、sequence中的一种作为主键生成方式，但是 OID必须为数据类型。

5）hilo：hilo标识符生成器由hibernate按照一种high/low（高地位）算法生成标识 符，他从数据库中的特定表的字段中获取high值。因此需要额外的数据表保 存主键生成的历史状态，hilo生成方法不依赖底层数据库，因此适用 于每一种数据库，但是OID必须为数值类型。

用法：

|  |
| --- |
| <id name=”id” type=”java.lang.Integer” column=”ID”>  <generator class=”hilo”>  <param name=”table”>my\_hi\_value</param>  <param name=”column”>next\_value</param>  </generator>  </id> |

6）uuid：由hibernate基于128位唯一值产生算法，根据当前设备IP，时间，JVM 启动时间，内部自增量等4个参数生成16进制数值作为主键，一般而言，利 用UUID方式生成的主键提供最好的数据插入性和数据库平台适应性。OID 一般使用String类型。

7）assigned：采用assign生成策略表示由应用程序逻辑来负责生成主键标识符，OID 类型没有限制。

8）映射复合主键：多个字段合起来作为一个主键。

9）foreign：一张表中一个字段既是本身的主键，又是一个外键指向另一张表。

在one-to-one的关系中，有另一张表的主键(Person) 来决定 自己主键/外 键( IdCard)。

1. 简单建议：

①针对Oracle：如果主键类型是int、short、long，建议使用sequence； 如果主键是String，建议使用uuid或者assigned。

②针对MySQL：如果主键是int、short、long，建议使用increment或者 assigned；如果主键是String，建议使用uuid或者assigned。

③针对SQLServer：如果主键是int、short、long，建议使用identity、native、 assigned；如果是String，建议使用uuid或者assigned。

Hibernate应用的场景：

1. 不适合OLAP（On-Line Analytical Processing联机分析处理），以查询分析数据为主的系统；
2. 适合OLTP（On-Line Transaction Processing联机事务处理）
3. 主要用于事务操作比较多的项目（OA/某个行业软件）
4. 数据量巨大，性能要求苛刻的系统，hibernate也很难达到要求，批量操作数据的效率也不高。