课程回顾：

1. 环境搭建

2. hibernate开发相关Api

3. hibernate.cfg.xml

4. \*.hbm.xml

主键字段映射

普通字段映射

5. 对象状态

6. lazy 懒加载

7. session缓存(一级缓存)

8. 一对多与多对一

9. 多对多

10. 一对一

11. 组件/继承映射

12. cascade关键字

目标：

1. 二级缓存使用

2. HQL查询

重点：连接查询

3. hibernate对连接池支持

4. session创建的2种方式

5. 案例

项目对session的管理方式的应用：

Open Session In View 模式

# 1. 二级缓存使用

Hibernate提供的缓存分类：

一级缓存

基于sessoin的缓存！

特点：

1. 在短时间内多次操作数据库情况下，缓存效果比较明显！

2. session关闭后，就不能使用缓存内容!

面试题：

不同的Session，能否共享一级缓存数据！

不能！

**实验：**

**1. 2个session**

**2. Dept dept = (Dept)session1.get(..);**

**dept加入session1的缓存**

**3. session2.update(dept);**

**dept 加入session2的缓存**

**4. dept.setName("new name");**

**总结：**

**缓存只在当前session有效，缓存时间短、作用范围小！总体来看缓存效果不明显！**

二级缓存:

基于应用程序的缓存、基于sessionFactory级别的缓存！

缓存数据可以被多个session共享! 但需要指定哪些对象要放入二级缓存中！

放入二级缓存中对象的特点：

1. 经常使用

2. 不会被经常修改！

Hibernate提供的二级缓存是以缓存框架形式提供，hibernate提供了二级缓存框架默认的实现； 也支持其他二级缓存框架，如果要更换缓存，只要更换配置中具体的二级缓存框架使用的核心类即可！ 可插配的缓存框架！

## 配置

Hibernate提供的二级缓存配置， hibenrate.properties配置文件

## disable the second-level cache 【二级缓存默认为关闭】

#hibernate.cache.use\_second\_level\_cache false

## enable the query cache 【是否开启查询缓存】

#hibernate.cache.use\_query\_cache true

## choose a cache implementation 【二级缓存的实现】

#hibernate.cache.provider\_class org.hibernate.cache.EhCacheProvider

#hibernate.cache.provider\_class org.hibernate.cache.EmptyCacheProvider

hibernate.cache.provider\_class org.hibernate.cache.HashtableCacheProvider

二级缓存默认的实现

#hibernate.cache.provider\_class org.hibernate.cache.TreeCacheProvider

#hibernate.cache.provider\_class org.hibernate.cache.OSCacheProvider

#hibernate.cache.provider\_class org.hibernate.cache.SwarmCacheProvider

## 步骤:

1) 开启二级缓存

2) 指定二级缓存具体实现框架

3) 那些类如要加入二级缓存

4) 测试

\* 2个session

\* session1.get(1); 查询数据库

Session1.get(); (没有发送查询sql)

\* session2.get(1); 从一级缓存中获取，没有，再找二级缓存，找到后就返回！

(没有发送查询sql, 说明应用成功！)

缓存策略:

usage="read-write" 二级缓存的数据可以读、写

usage="read-only" 二级缓存的数据只读

usage="nonstrict-read-write" 非严格读取

usage="transactional" 基于事务的策略

集合缓存

查询缓存

List()查询，不会从一级缓存获取？ 那二级缓存呢？

🡪 不会从session缓存获取数据

🡪 默认不会从二级缓存获取数据，但可以指定从二级缓存获取！

设计实验：

更新数据，会不会通知一级缓存、二级缓存？

不会通知一级缓存，会通知二级缓存！

1. 对象放入session缓存

2. 修改对象

3. 再看一级缓存数据有没有改变!

# 2. HQL 查询

hql语法基本与数据库查询语法一致！

支持：模糊、条件、分组、筛选、连接等！

HQL查询：

要有一定的sql基础！

# 3. Criteria 查询

Hibernate提供的面向对象的查询的另外的方式！

更面向对象！

# 3. SQLQuery 查询

Hibernate除了支持面向对象的查询外，还支持原生态的sql语句查询，在数据库段写好的sql也可以直接在hibernate中执行！

优缺点：

1 . 对于一些比较负责的查询， hql实现不了， 这时候，本地sql查询就可以作为补充！

2. 使用本地sql查询不能跨数据库平台！ 一旦更换了数据库，sql语句有可能会更改！

# 4. Hibernate 对连接池支持

连接池：

开源的连接池组件C3p0！

连接池：

管理连接，提高连接的使用效率！

Hibernate对连接的支持：

查看hibernate.properties

#################################

### Hibernate Connection Pool ###

#################################

hibernate.connection.pool\_size 1 hibernate自带的连接池，只有一个连接！

###########################

### C3P0 Connection Pool### hibernate对C3p0连接池的支持

###########################

#hibernate.c3p0.max\_size 2

#hibernate.c3p0.min\_size 2

#hibernate.c3p0.timeout 5000

#hibernate.c3p0.max\_statements 100

#hibernate.c3p0.idle\_test\_period 3000

#hibernate.c3p0.acquire\_increment 2

#hibernate.c3p0.validate false

C3p0连接池驱动类

#hibernate.connection.provider\_class org.hibernate.connection.C3P0ConnectionProvider

hibernate项目中使用c3p0连接池：

0. 引入c3p0驱动包

1. 配置c3p0驱动类

2. 连接池参数配置

# 5. 用户库/MyEclipse对hibernate的支持

MyEclipse支持以用户库来管理jar文件！

# 6. 创建Session的两种方式

|  |
| --- |
| // 2种方式创建session  @Test  **public** **void** testSession() **throws** Exception {  System.*out*.println(*sf*);  // 方式1: sf.openSession() 每次都创建一个新的session！  Session session1 = *sf*.openSession();  Session session2 = *sf*.openSession();  System.*out*.println(session1 == session2); // false  session1.close();  session2.close();    // 方式2： 线程的方式创建session  // getCurrentSession() 先从当前线程获取session，没有获取到，就创建新的session  // 创建完成后，再绑定到当前线程！  Session session3 = *sf*.getCurrentSession();  Session session4 = *sf*.getCurrentSession();  System.*out*.println(session3 == session4); // true  session3.close();  //session4.close(); // 不能重复关闭  } |

# 7. 项目案例

重点：

项目中对Session的管理！

步骤：

1. 环境搭建, 引入jar文件

Struts 相关jar

Hibernate相关jar

**注意：要检查用户库的jar文件，有没有发布到tomcat下！**

**如果没有，按照这里配置：《构建用户库-步骤图解.doc》**

2. 配置

web.xml

🡪 自己写的过滤器

🡪 struts核心过滤器

Struts.xml

Hibernate.cfg.xml

Dept.hbm.xml

Employee.hbm.xml

3. utils /filter/ dao/service/action

4. jsp

获取员工，以及关联的部门！