# 1.根据条件查找

sql = "from Users where name='"+users.getName()+"'and password = '"+users.getPassword()+"'";

# 2.查找一个表中的所有的条目

Query query=session.createQuery("from Student");

# 3.查找有限的字段

//检索学生的名字和有限字段

//原则上我们是 可以像jdbc中全部查找出来的

//hibernate 建议中都是必须都查出来

//但是我们也要学会 查找有限的

**private** **static** **void** selectOne(){

Session session = **null**;

Transaction ts = **null**;

**try** {

session = HibernateUtil.*getCurrentSession*();

ts = session.beginTransaction();

//检索所有学生的信息

Query query=session.createQuery("select sname,adept from Student");

//这样我们就不可以使用下面的东西了，就会报错因为哦我们不能得到一个list

List list=query.list();

**for**(**int** i = 0 ; i < list.size();i++){

Object []object = (Object[]) list.get(i);

System.*out*.println(object[0].toString()+""+object[1]);

}

ts.commit();

} **catch** (Exception e) {

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

ts.rollback();

}

**throw** **new** RuntimeException(e.getMessage()+"这个错误了");

}**finally**{

//关闭session

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

session.close();

}

}

}

# 4.范围查找20到23岁之间的

Query query=session.createQuery("from Student

where sage between 20 and 23");

# 5. distinct学生的性别和年龄，但是不能出现重复

Query query=session.createQuery("

select distinct ssex,sage from Student");

# 6. .uniqueResult();确认有一条数据，我们就是用

Student student = (Student)session.createQuery("from Student where sid = '1'").uniqueResult();

# 7. 查找包含某一类型的

List<Student> list=session.createQuery

("from Student where adept in ('computer')").list();

# 8.查找系里面的平均年龄

//下面这个是两个字段了 ，系里面的平均年龄

List<Object[]> list=session.createQuery("select avg(sage),adept from Student group by adept").list();

**for**(Object[] obj:list){

System.*out*.println(obj[0].toString()+" "+obj[1].toString());

}

# 9. //查询各个系有 人数 大于3 的系的名称 //查询这个系里有多少个学生

//count 用于别名 as c1

List<Object[]> list=session.createQuery("select count(\*) as c1,adept from Student group by adept having count(\*)>3").list();

/

**for**(Object[] obj:list){

System.*out*.println(obj[0].toString()+" "+obj[1].toString());

}

# 10//选择系里面男生有多少个 ，和系的名称

List<Object[]> list=session.

createQuery("select count(\*) as c1,adept from Student where ssex='m' group by adept").list();

//ȡ��1. for ��ǿ

**for**(Object[] obj:list){

System.*out*.println(obj[0].toString()+" "+obj[1].toString());

}

14

# 11主外键进行查找，查找课程为111的最大成绩和最小成绩

List<Object[]> list=session.

createQuery("select 111,max(grade),min(grade) from Stucourse where course.cid=111").list();

//ȡ��1. for ��ǿ

**for**(Object[] obj:list){

System.*out*.println(obj[0].toString()+" max="+obj[1].toString()+" min="+obj[2].toString());

}

## 12. 选择成绩 大于60的 学生的数目和所在的系

List<Object[]> list=session.

createQuery("select count(\*),student.adept from Stucourse where grade>60 group by student.adept").list();

//ȡ��1. for ��ǿ

**for**(Object[] obj:list){

System.*out*.println(obj[0].toString()+" "+obj[1].toString());

}