HQL:

1.分页查询：根据员工id，员工姓名（模糊），员工性别，所在部门查询员工信息并实现分页功能

DAO层将Hql语句写出,并预处理了查询条件（不在语句中的条件置为null，便于baseDao中的条件判断）,可以直接将pageResults对象返回。并且由于返回的是List<object[]>类型的列表，所以不能直接select e，需要将所要的列都写出来:select e.id,e.name,…

**public** List<Object[]> selectWithParamByPage(String id, String name,

String gender, String dept, **int** pageNo, **int** pageSize) {

// **TODO** Auto-generated method stub

StringBuilder hql = **new** StringBuilder(

"select e.id,e.name,e.gender,e.dept.name from EmplCopy as e where 1=1");

**if** (id != **null** && !"".equals(id)) {

hql.append(" and e.id=?");

} **else** {

id = **null**;

}

**if** (name != **null** && !"".equals(name)) {

hql.append(" and e.name like ?");

name = "%" + name + "%";

} **else** {

name = **null**;

}

**if** (gender != **null** && !"".equals(gender)) {

hql.append(" and e.gender=?");

} **else** {

gender = **null**;

}

**if** (dept != **null** && !"".equals(dept)) {

hql.append(" and e.dept.name=?");

} **else** {

dept = **null**;

}

hql.append(" order by e.id asc");

PageResults results = selectPageByFetchedHql(hql.toString(), **null**,

pageNo, pageSize, id, name, gender, dept);

System.*out*.println("当前页:" + results.getCurrentPage());

System.*out*.println("总页数" + results.getPageCount());

**return** results.getResults();

}

调用BaseDao中的

/\*\*

\* <HQL分页查询>

\*

\* **@param** hql

\* HQL语句

\* **@param** countHql

\* 查询记录条数的HQL语句

\* **@param** pageNo

\* 下一页

\* **@param** pageSize

\* 一页总条数

\* **@param** values

\* 不定Object数组参数

\* **@return** PageResults的封装类，里面包含了页码的信息以及查询的数据List集合

\* **@see** dao.IBaseDao#findPageByFetchedHql(java.lang.String,

\* java.lang.String, int, int, java.lang.Object[])

\*/

**public** PageResults selectPageByFetchedHql(String hql, String countHql,

**int** pageNo, **int** pageSize, Object... values) {

PageResults retValue = **new** PageResults();

Query query = **this**.getSession().createQuery(hql);

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

**int** temp = 0;

**for** (**int** i = 0; i < values.length; i++) {

**if** (values[i] != **null**)

query.setParameter(temp++, values[i]);

}

}

retValue.setCurrentPage(pageNo);

retValue.setPageSize(pageSize);

**if** (countHql == **null**) {

ScrollableResults results = query.scroll();

results.last();

retValue.setTotalCount(results.getRowNumber() + 1);// 设置总记录数

} **else** {

Long count = countByHql(countHql, values);

retValue.setTotalCount(count.intValue());

}

List<Object[]> itemList = query

.setFirstResult((retValue.getCurrentPage() - 1) \* pageSize)

.setMaxResults(pageSize).list();

**if** (itemList == **null**) {

itemList = **new** ArrayList<Object[]>();

}

retValue.setResults(itemList);

**return** retValue;

}

2. 获取员工id+姓名+性别+部门名称的组合字符串

select e.id||'+'||e.name||'+'||e.gender||'+'||e.dept.name from EmplCopy as e

3.获取所有员工详细信息

select e.id,e.name,e.gender,e.dept.name from EmplCopy as e

4.根据员工id更新：员工姓名，员工部门id（需要通过子查询）

这个是直接更新部门id

String hql = "update EmplCopy e set e.name=? , e.dept.id=? where e.id=?";

Query query = getSession().createQuery(hql);

query.setParameter(0, name);

query.setParameter(1, 5);

query.setParameter(2, id);

query.executeUpdate();

这个是嵌套了子查询：给的是员工的部门name

update EmplCopy e set e.name=? , e.dept.id=(select d.id from Department d where d.name=?) where e.id=?

5.向员工表中添加一条记录：对于员工部门，先查部门表，得出部门id，然后再对员工执行添加.

写一般的查询或更新语句使用getSession().createQuery();

所以baseDao中只需封装

1. selectById(ID id);

2.insert(T t)

3.delete(T t):没有返回值。删除时必须提供主键，否则不执行删除,并且只根据id判断，其他字段传入也不管用。下面是执行时打印的log

Hibernate: select emplcopy\_.id, emplcopy\_.deptId as deptId4\_1\_, emplcopy\_.gender as gender2\_1\_, emplcopy\_.name as name3\_1\_ from employee emplcopy\_ where emplcopy\_.id=?

Hibernate: delete from employee where id=?

4.deleteById(ID id):自定义，有返回值

5.deleteAll(Collection<T> entities);

6.update(T t):该方法会把传入的t中所有字段更新

当只更新其中几个字段而非全部字段时，更新操作：

1. 使用query.ExecuteUpdate方法执行
2. 如果有主键的话，首先执行selectById方法获取数据库中的实体

然后将相应要改变的值修改到该对象中，然后调用update方法

7.分页方法

首先拼接HQL查询语句，将无效的查询参数置空。然后调用该分页方法。