# HQL检索方式

## List查询

要注意的是from后面接的不是表名，而是你定义的那个实体类名，否则会出现not mapped错误！！

|  |
| --- |
| Transaction tx = session.beginTransaction();  Query q = session.createQuery("from Teacher where name=:v\_name");  q.setString("v\_name", "huang");  List<Teacher> l = q.list();  Iterator i = l.iterator();  **while**(i.hasNext()){  Teacher t = (Teacher)i.next();  System.***out***.println(t.getId()+t.getName()+t.getTitle());  } |

这个是不带条件的查询，把表里面所有的数据都弄出来

|  |
| --- |
| Transaction tx = session.beginTransaction();  String hql = "from Teacher";  Query q = session.createQuery(hql);  //q.setString("v\_name", "huang");  List<Teacher> l = q.list();  Iterator i = l.iterator();  **while**(i.hasNext()){  Teacher t = (Teacher)i.next();  System.***out***.println(t.getId()+t.getName()+t.getTitle());  } |

List查询方式，会占用很大的空间，容易造成资源的浪费

## Iterator查询

这个相当于数据库中的游标，相当于指针

|  |
| --- |
| Transaction tx = session.beginTransaction();  String hql = "from Teacher";  Query q = session.createQuery(hql);  Iterator it = q.iterate();  **while**(it.hasNext()){  Teacher t = (Teacher)it.next();  System.***out***.println(t.getId()+t.getName()+t.getTitle());  } |

## Query可以完成更新/删除操作

|  |
| --- |
| Transaction tx = session.beginTransaction();  String hql = "update Teacher set name='ning' where id=3";  //String hql = “delete from ..”  Query q = session.createQuery(hql);  q.executeUpdate();  tx.commit(); |

# Critiria接口查询

## 例子

|  |
| --- |
| Transaction tx = session.beginTransaction();  Criteria c = session.createCriteria(Teacher.**class**);  c.add(Restrictions.*eq*("name", "huang"));  List<Teacher> l = c.list();  Iterator i = l.iterator();  **while**(i.hasNext()){  Teacher t = (Teacher)i.next();  System.***out***.println(t.getId()+t.getName()+t.getTitle());  }  tx.commit(); |

彻底抛弃sql语句了，查询条件被封装成了Criteria对象了