

	HQL	Criteria
查询所有	<pre>String hql = "from Employee"; List<Employee> result = (List<Employee>)session.createQuery(hql) .list();</pre>	<pre>List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .list();</pre>
条件过滤 =、!= >、>= <、<=	<pre>String hql = "from Employee e where e.age = :age"; List<Employee> result = (List<Employee>)session.createQuery(hql) .setInteger("age", age) .list();</pre>	<pre>List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .add(Restrictions.eq("age", age)) // .add(Restrictions.ne("age", age)) // .add(Restrictions.gt("age", age)) // .add(Restrictions.ge("age", age)) // .add(Restrictions.lt("age", age)) // .add(Restrictions.le("age", age)) .list();</pre>
条件过滤 between and	<pre>String hql = "from Employee e where e.age between :fromAge and :toAge"; List<Employee> result = (List<Employee>)session.createQuery(hql) .setInteger("fromAge", fromAge) .setInteger("toAge", toAge) .list();</pre>	<pre>List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .add(Restrictions.between("age", fromAge, toAge)) .list();</pre>
条件过滤 is null	<pre>String hql = "from Employee e where e.department is null"; //String hql = "from Employee e where e.department is not null"; List<Employee> result =</pre>	<pre>List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .add(Restrictions.isNull("department")) // .add(Restrictions.isNotNull("department")) .list();</pre>

	<pre>(List<Employee>)session.createQuery(hql) .list();</pre>	
条件过滤 集合属性 size	<pre>//String hql = "from Department d where d.employees is empty"; String hql = "from Department d where <u>d.employees is not empty</u>"; String hql = "from Department d where d.employees.size > 5"; String hql = "from Department d where d.employees.size < 5"; List<Department> result = (List<Department>)session.createQuery(hql) .list();</pre>	<pre>List<Department> result = (List<Department>)session.createCriteria(Department.cla ss) // .add(Restrictions.isEmpty("employees")) <u>.add(Restrictions.isNotEmpty("employees"))</u> // .add(Restrictions.sizeGt("employees", 5)) // .add(Restrictions.sizeLt("employees", 5)) .list();</pre>
条件过滤 通配符	<pre>String hql = "from Employee e where <u>e.name like :name</u>"; List<Employee> result = (List<Employee>)session.createQuery(hql) .setString("name", name+"%") .list();</pre>	<pre>List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .add(Restrictions.like("name", name + "%")) // .add(Restrictions.like("name", name, MatchMode.START)) .list();</pre>
条件过滤 多条件	<pre>String hql = "from Employee e where <u>(e.department is not null and e.age > 25) or e.age < 25</u>"; List<Employee> result = (List<Employee>)session.createQuery(hql) .list();</pre>	<pre>List<Employee> result = (List<Employee>) session.createCriteria(Employee.class) .add(<u>Restrictions.or(Restrictions.and(Restrictions.isNotNull("department"), Restrictions.gt("age", 25)),</u></pre>

		<u>Restrictions.lt("age", 25))</u> <u>)</u> .list();
条件过滤 in	String hql = "from Employee e where <u>e.id in (1, 2, 3)</u> "; List<Employee> result = (List<Employee>)session.createQuery(hql) .list();	List<Employee> result = (List<Employee>) session.createCriteria(Employee. class) . <u>add(Restrictions.in("id", new Integer[]{1,2,3}))</u> .list();
排序	String hql ="from Employee e <u>order by e.age</u> "; String hql ="from Employee e order by e.age desc";	session.createCriteria(Employee. class) . <u>addOrder(Order.asc("age"))</u> .addOrder(Order.desc("age"))
投影 select 1 列	String hql = "select <u>e.name</u> from Employee e"; Query query = session.createQuery(hql); List<Object> result = (List<Object>)query.list();	List<Object> result = (List<Object>)session.createCriteria(Employee. class) . <u>setProjection(Projections.property("name"))</u> .list();
投影 select 多列	String hql = "select e.name, e.age from Employee e"; Query query = session.createQuery(hql); List<Object[]> result = (List<Object[]>)query.list();	List<Object[]> result = (List<Object[]>)session.createCriteria(Employee. class) . <u>setProjection(Projections.projectionList()</u> <u>.add(Projections.property("name"))</u> <u>.add(Projections.property("age"))</u> <u>)</u> .list();
聚集 Count Max Min	String hql = "select <u>count(age), max(age), min(age), avg(age), sum(age)</u> from Employee"; Object[] result =	Object[] result = (Object[]) session.createCriteria(Employee. class) . <u>setProjection(Projections.projectionList()</u> <u>.add(Projections.rowCount())</u>

Avg Sum	<pre>(Object[])session.createQuery(hql) .uniqueResult();</pre>	<pre>.add(Projections.max("age")) .add(Projections.min("age")) .add(Projections.avg("age")) .add(Projections.sum("age"))) .uniqueResult();</pre>
投影 Result Transfer	<pre>String hql = "select new com.jeff.vo.StatInfo(count(e.id), max(e.age), min(e.age), avg(e.age)) from Employee e"; Query query = session.createQuery(hql); List<StatInfo> result = (List<StatInfo>)query.list();</pre>	<pre>List<StatInfo> result = (List<StatInfo>)session.createCriteria(Employee.class) .setProjection(Projections.projectionList() .add(Projections.count("id").as("count")) .add(Projections.max("age").as("maxAge")) .add(Projections.min("age").as("minAge")) .add(Projections.avg("age").as("avgAge"))) .setResultTransformer(Transformers.aliasToBean(StatI nfo.class)) .list();</pre>
分组 Group by	<pre>String hql = "select e.department.name, count(age), max(age), min(age), avg(age), sum(age) from Employee e group by e.department"; List<Object[]> result = (List<Object[]>)session.createQuery(hql) .list();</pre>	<pre>List<Object[]> result = (List<Object[]>)session.createCriteria(Employee.class) .setProjection(Projections.projectionList() .add(Projections.property("department")) .add(Projections.rowCount()) .add(Projections.max("age")) .add(Projections.min("age")) .add(Projections.avg("age"))</pre>

		<pre> .add(Projections.sum("age")) .add(Projections.groupProperty("department"))) .list(); </pre>
联结 Join	<pre> //String hql = "from Employee e where e.department.id = 9"; //隐式联结 String hql = "select e from Employee e join e.department d where d.id = 9"; //显式联结 Query query = session.createQuery(hql); List<Employee> result = (List<Employee>)query.list(); </pre>	<pre> List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) //createCriteria是Session的方法 .add(Restrictions.gt("age", 25)) .createCriteria("department") //createCriteria是 Criteria的方法 .add(Restrictions.eq("id", 9)) .list(); List<Employee> result = (List<Employee>)session.createCriteria(Employee.class) .createAlias("department", "d") .add(Restrictions.gt("age", 25)) .add(Restrictions.eq("d.id", 9)) .list(); </pre>
动态抓取 Fetch	<pre> String hql = "from Department d left join fetch d.employees where d.id = 9"; Query query = session.createQuery(hql); department = (Department)query.uniqueResult(); </pre>	<pre> department = (Department)session.createCriteria(Department.class) .setFetchMode("employees", FetchMode.JOIN) .add(Restrictions.eq("id", 9)) .uniqueResult(); </pre>
命名查询 Named	<pre> //Query query = session.getNamedQuery("getByAge"); </pre>	

Query	<pre> Query query = session.<u>getNamedQuery</u>("com.jeff.model.Employee .getByAge"); query.setInteger("ageParam", 25); List<Employee> result = (List<Employee>)query.list(); </pre>	
离线查询 Detache dCriter ia		<pre> DetachedCriteria cond = DetachedCriteria.<u>forClass</u>(Employee.class) .add(Restrictions.eq("age", 25)); List<Employee> result = cond.<u>getExecutableCriteria</u>(session).list(); </pre>