**[Hibernate分页查询小结](http://blog.csdn.net/suntao1983/article/details/3607835)**

通常使用的Hibernate通常是三种：hql查询，QBC查询和QBE查询：   
1、QBE（Qurey By Example）检索方式   
QBE 是最简单的，但是功能也是最弱的，QBE的功能不是特别强大，仅在某些场合下有用。一个典型的使用场合就是在查询窗口中让用户输入一系列的查询条件，然后 返回匹配的对象。QBE只支持=和like比较运算符，无法不大区间值，及其或的匹配。在这种情况下，还是采用HQL检索方式或QBC检索方式。

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 根据传递过来的Object,分页显示在数据库中与其匹配的记录
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param object
8. \*            将查询条件封装为Object
9. \* @return 将查询结果封装为Pager返回
10. \*/
11. **public** Pager findPageByExample(**int** pageNo, **int** pageSize, Object object)
12. {
13. Pager pager = **null**;
14. **try**
15. {
16. Criteria criteria = **this**.getSession().createCriteria(
17. Class.forName(**this**.getEntity()));
19. **if** (object != **null**)
20. {
21. criteria.add(Example.create(object).enableLike());
22. }
24. // 获取根据条件分页查询的总行数
25. **int** rowCount = (Integer) criteria.setProjection(
26. Projections.rowCount()).uniqueResult();
27. criteria.setProjection(**null**);
29. criteria.setFirstResult((pageNo - 1) \* pageSize);
30. criteria.setMaxResults(pageSize);
32. List result = criteria.list();
34. pager = **new** Pager(pageSize, pageNo, rowCount, result);
36. } **catch** (RuntimeException re)
37. {
38. **throw** re;
39. } **finally**
40. {
41. **return** pager;
42. }
44. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 根据传递过来的Object,分页显示在数据库中与其匹配的记录
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param object
8. \*            将查询条件封装为Object
9. \* @return 将查询结果封装为Pager返回
10. \*/
11. public Pager findPageByExample(int pageNo, int pageSize, Object object)
12. {
13. Pager pager = null;
14. try
15. {
16. Criteria criteria = this.getSession().createCriteria(
17. Class.forName(this.getEntity()));
19. if (object != null)
20. {
21. criteria.add(Example.create(object).enableLike());
22. }
24. // 获取根据条件分页查询的总行数
25. int rowCount = (Integer) criteria.setProjection(
26. Projections.rowCount()).uniqueResult();
27. criteria.setProjection(null);
29. criteria.setFirstResult((pageNo - 1) \* pageSize);
30. criteria.setMaxResults(pageSize);
32. List result = criteria.list();
34. pager = new Pager(pageSize, pageNo, rowCount, result);
36. } catch (RuntimeException re)
37. {
38. throw re;
39. } finally
40. {
41. return pager;
42. }
44. }

注意代码的第20行，即criteria.add(Example.create(object).enableLike());这一行，需将Example.create(object)调用.enableLike()方法，不然不能模糊查询。   
在BO层将需要模糊查询的列用"%%"串起来，不然仍然和"="一样。   
BO层代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. **public** Pager getInfoByQuery(**int** pageNo, **int** pageSize, String mendName,
16. String specialty, String post)
17. {
19. EicMend eicMend = **new** EicMend();
20. **if** (mendName != **null** && mendName.length() > 0)
21. {
22. eicMend.setMendname("%" + mendName + "%");
23. }
24. **if** (specialty != **null** && specialty.length() > 0)
25. {
26. eicMend.setSpecialty(specialty);
27. }
28. **if** (post != **null** && post.length() > 0)
29. {
30. eicMend.setPost(post);
31. }
33. Pager pager = erpManagerDao
34. .findPageByExample(pageNo, pageSize, eicMend);
35. **return** pager;
36. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. public Pager getInfoByQuery(int pageNo, int pageSize, String mendName,
16. String specialty, String post)
17. {
19. EicMend eicMend = new EicMend();
20. if (mendName != null && mendName.length() > 0)
21. {
22. eicMend.setMendname("%" + mendName + "%");
23. }
24. if (specialty != null && specialty.length() > 0)
25. {
26. eicMend.setSpecialty(specialty);
27. }
28. if (post != null && post.length() > 0)
29. {
30. eicMend.setPost(post);
31. }
33. Pager pager = erpManagerDao
34. .findPageByExample(pageNo, pageSize, eicMend);
35. return pager;
36. }

执行SQL语句如下：

Sql代码 [复制代码](http://www.javaeye.com/topic/266301)

1. Hibernate: **select** count(\*) **as** y0\_ **from** YJZX.EIC\_MEND this\_ **where**
2. (this\_.MENDNAME like ? and this\_.POST like ?)
4. Hibernate: **select** \* **from** ( **select** this\_.MENDID **as** MENDID23\_0\_, ……
5. this\_.EXPERTREMARK **as** EXPERTR28\_23\_0\_ **from** YJZX.EIC\_MEND this\_ **where**
6. (this\_.MENDNAME like ? and this\_.POST like ?) ) **where** rownum <= ?

**[sql]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. Hibernate: select count(\*) as y0\_ from YJZX.EIC\_MEND this\_ where
2. (this\_.MENDNAME like ? and this\_.POST like ?)
4. Hibernate: select \* from ( select this\_.MENDID as MENDID23\_0\_, ……
5. this\_.EXPERTREMARK as EXPERTR28\_23\_0\_ from YJZX.EIC\_MEND this\_ where
6. (this\_.MENDNAME like ? and this\_.POST like ?) ) where rownum <= ?

所以只需将需模糊查询的列用“%%”链接即可。   
  
2、QBC（Qurey By Criteria）检索方式   
       采用HQL检索方式时，在应用程序中需要定义基于字符串形式的HQL查询语句。QBC API提供了检索对象的另一种方式，它主要由Criteria接口、Criterion接口和Restrictions接口组成，它支持在运行时动态生成 查询语句。比较常见的是两种传参方式：一种是用map传参，另一种是用Criterion…不定参数传参。   
Map传参方式范例如下：   
DAO层：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param map
8. \*            将查询条件封装为map
9. \* @return 查询结果Pager
10. \*/
11. **public** Pager findPageByCriteria(**int** pageNo, **int** pageSize, Map map)
12. {
13. Pager pager = **null**;
14. **try**
15. {
16. Criteria criteria = **this**.getSession().createCriteria(
17. Class.forName(**this**.getEntity()));
19. **if** (map != **null**)
20. {
21. Set<String> keys = map.keySet();
22. **for** (String key : keys)
23. {
24. criteria.add(Restrictions.like(key, map.get(key)));
25. }
26. }
28. // 获取根据条件分页查询的总行数
29. **int** rowCount = (Integer) criteria.setProjection(
30. Projections.rowCount()).uniqueResult();
31. criteria.setProjection(**null**);
33. criteria.setFirstResult((pageNo - 1) \* pageSize);
34. criteria.setMaxResults(pageSize);
36. List result = criteria.list();
38. pager = **new** Pager(pageSize, pageNo, rowCount, result);
40. } **catch** (RuntimeException re)
41. {
42. **throw** re;
43. } **finally**
44. {
45. **return** pager;
46. }
48. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param map
8. \*            将查询条件封装为map
9. \* @return 查询结果Pager
10. \*/
11. public Pager findPageByCriteria(int pageNo, int pageSize, Map map)
12. {
13. Pager pager = null;
14. try
15. {
16. Criteria criteria = this.getSession().createCriteria(
17. Class.forName(this.getEntity()));
19. if (map != null)
20. {
21. Set<String> keys = map.keySet();
22. for (String key : keys)
23. {
24. criteria.add(Restrictions.like(key, map.get(key)));
25. }
26. }
28. // 获取根据条件分页查询的总行数
29. int rowCount = (Integer) criteria.setProjection(
30. Projections.rowCount()).uniqueResult();
31. criteria.setProjection(null);
33. criteria.setFirstResult((pageNo - 1) \* pageSize);
34. criteria.setMaxResults(pageSize);
36. List result = criteria.list();
38. pager = new Pager(pageSize, pageNo, rowCount, result);
40. } catch (RuntimeException re)
41. {
42. throw re;
43. } finally
44. {
45. return pager;
46. }
48. }

Map传参方式对应BO层代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. **public** Pager getInfoByQuery2(**int** pageNo, **int** pageSize, String mendName,
16. String specialty, String post)
17. {
19. Map map = **new** HashMap();
21. **if** (mendName != **null** && mendName.length() > 0)
22. {
23. map.put("mendname", "%" + mendName + "%");
24. }
25. **if** (specialty != **null** && specialty.length() > 0)
26. {
27. map.put("specialty", specialty);
28. }
29. **if** (post != **null** && post.length() > 0)
30. {
31. map.put("post", post);
32. }
34. Pager pager = erpManagerDao.findPageByCriteria(pageNo, pageSize, map);
35. **return** pager;
36. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. public Pager getInfoByQuery2(int pageNo, int pageSize, String mendName,
16. String specialty, String post)
17. {
19. Map map = new HashMap();
21. if (mendName != null && mendName.length() > 0)
22. {
23. map.put("mendname", "%" + mendName + "%");
24. }
25. if (specialty != null && specialty.length() > 0)
26. {
27. map.put("specialty", specialty);
28. }
29. if (post != null && post.length() > 0)
30. {
31. map.put("post", post);
32. }
34. Pager pager = erpManagerDao.findPageByCriteria(pageNo, pageSize, map);
35. return pager;
36. }

第二种方式：Criterion…不定参数传参方式。其代码如下所示：   
DAO层代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param criterions
8. \*            不定参数Criterion
9. \* @return 查询结果Pager
10. \*/
11. **public** Pager findPageByCriteria(**int** pageNo, **int** pageSize,
12. Criterion... criterions)
13. {
14. Pager pager = **null**;
15. **try**
16. {
17. Criteria criteria = **this**.getSession().createCriteria(
18. Class.forName(**this**.getEntity()));
19. **if** (criterions != **null**)
20. {
21. **for** (Criterion criterion : criterions)
22. {
23. **if** (criterion != **null**)
24. {
25. criteria.add(criterion);
26. }
28. }
29. }
31. // 获取根据条件分页查询的总行数
32. **int** rowCount = (Integer) criteria.setProjection(
33. Projections.rowCount()).uniqueResult();
34. criteria.setProjection(**null**);
36. criteria.setFirstResult((pageNo - 1) \* pageSize);
37. criteria.setMaxResults(pageSize);
39. List result = criteria.list();
41. pager = **new** Pager(pageSize, pageNo, rowCount, result);
43. } **catch** (RuntimeException re)
44. {
45. **throw** re;
46. } **finally**
47. {
48. **return** pager;
49. }
51. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param criterions
8. \*            不定参数Criterion
9. \* @return 查询结果Pager
10. \*/
11. public Pager findPageByCriteria(int pageNo, int pageSize,
12. Criterion... criterions)
13. {
14. Pager pager = null;
15. try
16. {
17. Criteria criteria = this.getSession().createCriteria(
18. Class.forName(this.getEntity()));
19. if (criterions != null)
20. {
21. for (Criterion criterion : criterions)
22. {
23. if (criterion != null)
24. {
25. criteria.add(criterion);
26. }
28. }
29. }
31. // 获取根据条件分页查询的总行数
32. int rowCount = (Integer) criteria.setProjection(
33. Projections.rowCount()).uniqueResult();
34. criteria.setProjection(null);
36. criteria.setFirstResult((pageNo - 1) \* pageSize);
37. criteria.setMaxResults(pageSize);
39. List result = criteria.list();
41. pager = new Pager(pageSize, pageNo, rowCount, result);
43. } catch (RuntimeException re)
44. {
45. throw re;
46. } finally
47. {
48. return pager;
49. }
51. }

Criterion…不定参数传参方式对应BO层代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. **public** Pager getInfoByQuery3(**int** pageNo, **int** pageSize, String mendName,
16. String specialty, String post)
17. {
18. Criterion criterion1 = **null**, criterion2 = **null**, criterion3 = **null**;
19. **if** (mendName != **null** && mendName.length() > 0)
20. {
21. criterion1 = Restrictions.ilike("mendname", mendName,
22. MatchMode.ANYWHERE);
23. }
25. **if** (specialty != **null** && specialty.length() > 0)
26. {
27. criterion2 = Restrictions.ilike("specialty", specialty,
28. MatchMode.EXACT);
29. }
31. **if** (post != **null** && post.length() > 0)
32. {
33. criterion3 = Restrictions.ilike("post", post, MatchMode.EXACT);
34. }
36. Pager pager = erpManagerDao.findPageByCriteria(pageNo, pageSize,
37. criterion1, criterion2, criterion3);
39. **return** pager;
40. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 将传递过来的参数封装成抢修人员Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param mendName
8. \*            抢修人员的名称
9. \* @param specialty
10. \*            抢修人员的工种
11. \* @param post
12. \*            抢修人员的职称
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. public Pager getInfoByQuery3(int pageNo, int pageSize, String mendName,
16. String specialty, String post)
17. {
18. Criterion criterion1 = null, criterion2 = null, criterion3 = null;
19. if (mendName != null && mendName.length() > 0)
20. {
21. criterion1 = Restrictions.ilike("mendname", mendName,
22. MatchMode.ANYWHERE);
23. }
25. if (specialty != null && specialty.length() > 0)
26. {
27. criterion2 = Restrictions.ilike("specialty", specialty,
28. MatchMode.EXACT);
29. }
31. if (post != null && post.length() > 0)
32. {
33. criterion3 = Restrictions.ilike("post", post, MatchMode.EXACT);
34. }
36. Pager pager = erpManagerDao.findPageByCriteria(pageNo, pageSize,
37. criterion1, criterion2, criterion3);
39. return pager;
40. }

3、HQL检索方式   
  
HQL（Hibernate Query Language）是面向对象的查询语言，它和SQL查询语言有些相识。在Hibernate提供的各种检索方式中，HQL是使用最广的一种检索方式。   
使用Query接口分页查询DAO代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param instance
8. \*            将查询条件封装为专家Bean
9. \* @return 查询结果Pager
10. \*/
11. **public** List<Object> findPageByQuery(**int** pageNo, **int** pageSize, String hql,
12. Map map)
13. {
14. List<Object> result = **null**;
15. **try**
16. {
17. Query query = **this**.getSession().createQuery(hql);
19. Iterator it = map.keySet().iterator();
20. **while** (it.hasNext())
21. {
22. Object key = it.next();
23. query.setParameter(key.toString(), map.get(key));
24. }
26. query.setFirstResult((pageNo - 1) \* pageSize);
27. query.setMaxResults(pageSize);
29. result = query.list();
31. } **catch** (RuntimeException re)
32. {
33. **throw** re;
34. }
35. **return** result;
36. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 分页显示符合所有的记录数，将查询结果封装为Pager
3. \* @param pageNo
4. \*            当前页数
5. \* @param pageSize
6. \*            每页显示的条数
7. \* @param instance
8. \*            将查询条件封装为专家Bean
9. \* @return 查询结果Pager
10. \*/
11. public List<Object> findPageByQuery(int pageNo, int pageSize, String hql,
12. Map map)
13. {
14. List<Object> result = null;
15. try
16. {
17. Query query = this.getSession().createQuery(hql);
19. Iterator it = map.keySet().iterator();
20. while (it.hasNext())
21. {
22. Object key = it.next();
23. query.setParameter(key.toString(), map.get(key));
24. }
26. query.setFirstResult((pageNo - 1) \* pageSize);
27. query.setMaxResults(pageSize);
29. result = query.list();
31. } catch (RuntimeException re)
32. {
33. throw re;
34. }
35. return result;
36. }

查询所有记录数的DAO代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 根据查询条件查询记录数的个数
3. \* @param hql
4. \*            hql查询语句
5. \* @param map
6. \*            用map封装查询条件
7. \* @return 数据库中满足查询条件的数据的条数
8. \*/
9. **public** **int** getTotalCount(String hql, Map map)
10. {
11. **try**
12. {
13. Query query = **this**.getSession().createQuery(hql);
15. Iterator it = map.keySet().iterator();
16. **while** (it.hasNext())
17. {
18. Object key = it.next();
19. query.setParameter(key.toString(), map.get(key));
20. }
22. Integer i = (Integer) query.list().get(0);
23. **return** i;
24. } **catch** (RuntimeException re)
25. {
26. **throw** re;
27. }
29. }

**[java]** [view plaincopy](http://blog.csdn.net/suntao1983/article/details/3607835)

1. /\*\*
2. \* @function 根据查询条件查询记录数的个数
3. \* @param hql
4. \*            hql查询语句
5. \* @param map
6. \*            用map封装查询条件
7. \* @return 数据库中满足查询条件的数据的条数
8. \*/
9. public int getTotalCount(String hql, Map map)
10. {
11. try
12. {
13. Query query = this.getSession().createQuery(hql);
15. Iterator it = map.keySet().iterator();
16. while (it.hasNext())
17. {
18. Object key = it.next();
19. query.setParameter(key.toString(), map.get(key));
20. }
22. Integer i = (Integer) query.list().get(0);
23. return i;
24. } catch (RuntimeException re)
25. {
26. throw re;
27. }
29. }

BO层代码：

Java代码 [复制代码](http://www.javaeye.com/topic/266301)

1. /\*\*
2. \* @function 将传递过来的参数封装成专家Bean，分页查询符合条件的记录
3. \* @param pageNo
4. \*            当前的页码
5. \* @param pageSize
6. \*            每页显示的记录数
7. \* @param expertName
8. \*            专家的名称
9. \* @param expertSpecialty
10. \*            专家的专业类别
11. \* @param post
12. \*            专家的行政职位
13. \* @return 将符合条件的记录数以及页码信息封装成PagerBean返回
14. \*/
15. **public** Pager getInfoByQuery(**int** pageNo, **int** pageSize, String expertName,
16. String expertSpecialty, String post)
17. {
18. StringBuffer hql = **new** StringBuffer();
19. hql.append("select count(expertid) from EicExpert where 1=1 ");
21. Map map = **new** HashMap();
23. **if** (expertName != **null** && expertName.length() > 0)
24. {
25. map.put("expertname", "%" + expertName + "%");
26. hql.append("and expertname like :expertname ");
27. }
28. **if** (expertSpecialty != **null** && expertSpecialty.length() > 0)
29. {
30. map.put("expertspecialty", expertSpecialty);
31. hql.append("and expertspecialty like :expertspecialty ");
32. }
33. **if** (post != **null** && post.length() > 0)
34. {
35. map.put("post", post);
36. hql.append("and post like :post ");
37. }
39. String queryHql = hql.substring(22);
40. List result = erpManagerDao.findPageByQuery(pageNo, pageSize,
41. queryHql, map);
42. **int** rowCount = erpManagerDao.getTotalCount(hql.toString(), map);
44. Pager pager = **new** Pager(pageSize, pageNo, rowCount, result);
45. **return** pager;
46. }