1. 查询整个映射对象所有字段

**Java代码  IMG_256**

1. //直接from查询出来的是一个映射对象，即：查询整个映射对象所有字段
2. String hql = "from Users";
3. Query query = session.createQuery(hql);
5. List<Users> users = query.list();
6. **for**(Users user : users){
7. System.out.println(user.getName() + " : " + user.getPasswd() + " : " + user.getId());
8. }
10. 输出结果为：
11. name1 : password1 : 1
12. name2 : password2 : 2
13. name3 : password3 : 3

2.查询单个字段

**Java代码  IMG_257**

1. //查询单个字段
2. String hql = " select name from Users";
3. Query query = session.createQuery(hql);
5. List<String> list = query.list();
6. **for**(String str : list){
7. System.out.println(str);
8. }
9. 输出结果为：
10. name1
11. name2
12. name3

3.查询其中几个字段 

**Java代码  IMG_258**

1. //查询其中几个字段
2. String hql = " select name,passwd from Users";
3. Query query = session.createQuery(hql);
4. //默认查询出来的list里存放的是一个Object数组
5. List<Object[]> list = query.list();
6. **for**(Object[] object : list){
7. String name = (String)object[0];
8. String passwd = (String)object[1];
10. System.out.println(name + " : " + passwd);
11. }
12. 输出结果为：
13. name1 : password1
14. name2 : password2
15. name3 : password3

4.修改默认查询结果(query.list())不以Object[]数组形式返回，以List形式返回

**引用**

//查询其中几个字段,添加new list(),注意list里的l是小写的。也不需要导入包，这样通过query.list()出来的list里存放的不再是默认的Object数组了，而是List集合了     
        String hql = " select new list(name,passwd) from Users";     
        Query query = session.createQuery(hql);     
        //默认查询出来的list里存放的是一个Object数组，但是在这里list里存放的不再是默认的Object数组了，而是List集合了     
        List<List> list = query.list();     
        for(List user : list){     
            String name = (String)user.get(0);     
            String passwd = (String)user.get(1);     
                 
            System.out.println(name + " : " + passwd);     
        }     
        /\*\*    
        输出结果为：    
         name1 : password1    
        name2 : password2    
        name3 : password3    
         \*/

5.修改默认查询结果(query.list())不以Object[]数组形式返回，以Map形式返回

**Java代码  IMG_259**

1. //查询其中几个字段,添加new map(),注意map里的m是小写的。也不需要导入包，这样通过query.list()出来的list里存放的不再是默认的Object数组了，而是map集合了
2. String hql = " select new map(name,passwd) from Users";
3. Query query = session.createQuery(hql);
4. //默认查询出来的list里存放的是一个Object数组，但是在这里list里存放的不再是默认的Object数组了，而是Map集合了
5. List<Map> list = query.list();
6. **for**(Map user : list){
7. //一条记录里所有的字段值都是map里的一个元素,key是字符串0,1,2,3....，value是字段值
8. //如果将hql改为：String hql = " select new map(name as username,passwd as password) from Users";,那么key将不是字符串0,1,2...了，而是"username","password"了
9. String name = (String)user.get("0");//get("0");是get(key),注意:0,1,2...是字符串，而不是整形
10. String passwd = (String)user.get("1");
12. System.out.println(name + " : " + passwd);
13. }
14. /\*\*
15. 输出结果为：
16. name1 : password1
17. name2 : password2
18. name3 : password3
19. \*/

6.修改默认查询结果(query.list())不以Object[]数组形式返回，以Set形式返回,但是因为Set里是不允许有重复的元素，所以:username和password的值不能相同。只需将hql改为：String hql = " select new set(name,passwd) from Users";   
  
7.修改默认查询结果(query.list())不以Object[]数组形式返回，以自定义类型返回   
  
自定义类： 

**Java代码  IMG_260**

1. **package** com.domain;
3. **public** **class** MyUser {
5. **private** String username;
6. **private** String password;
7. //因为：String hql = " select new  com.domain.MyUser(name,passwd) from Users";所以必须要有接受2个参数的构造函数
8. **public** MyUser(String username,String password){
9. **this**.username = username;
10. **this**.password = password;
11. }
13. **public** String getUsername() {
14. **return** username;
15. }
16. **public** **void** setUsername(String username) {
17. **this**.username = username;
18. }
19. **public** String getPassword() {
20. **return** password;
21. }
22. **public** **void** setPassword(String password) {
23. **this**.password = password;
24. }

27. }

**Java代码  IMG_261**

1. //通过query.list()出来的list里存放的不再是默认的Object数组了,而是自定义的类MyUser,必须加包名，String hql = "from Users";中的Users类也是必须加包名的，但是因为再Users.hbm.xml里<hibernate-mapping auto-import="true"> auto-import默认值为true（所以auto-import属性也可以不写）,自动导入了
2. String hql = " select new  com.domain.MyUser(name,passwd) from Users";
3. Query query = session.createQuery(hql);
4. //默认查询出来的list里存放的是一个Object数组，但是在这里list里存放的不再是默认的Object数组了，而是MyUser对象了
5. List<MyUser> myUsers = query.list();
6. **for**(MyUser myUser : myUsers){
7. String name = myUser.getUsername();
8. String passwd = myUser.getPassword();
9. System.out.println(name + " : " + passwd);
10. }
11. /\*\*
12. 输出结果为：
13. name1 : password1
14. name2 : password2
15. name3 : password3
16. \*/

8：条件查询

**Java代码  IMG_262**

1. //条件查询，参数索引值从0开始，索引位置。通过setString,setParameter设置参数
2. String hql = "from Users where name=? and passwd=?";
3. Query query = session.createQuery(hql);
4. //第1种方式
5. //      query.setString(0, "name1");
6. //      query.setString(1, "password1");
7. //第2种方式
8. query.setParameter(0, "name1",Hibernate.STRING);
9. query.setParameter(1, "password1",Hibernate.STRING);
10. List<Users> list = query.list();
11. **for**(Users users : list){
12. System.out.println(users.getId());
13. }

**Java代码  IMG_263**

1. //条件查询，自定义索引名(参数名):username,:password.通过setString,setParameter设置参数
2. String hql = "from Users where name=:username and passwd=:password";
3. Query query = session.createQuery(hql);
4. //第1种方式
5. //      query.setString("username", "name1");
6. //      query.setString("password", "password1");
7. //第2种方式,第3个参数确定类型
8. query.setParameter("username", "name1",Hibernate.STRING);
9. query.setParameter("password", "password1",Hibernate.STRING);
10. List<Users> list = query.list();
11. **for**(Users users : list){
12. System.out.println(users.getId());
13. }

**Java代码  IMG_264**

1. //条件查询,通过setProperties设置参数
2. String hql = "from Users where name=:username and passwd=:password";
3. Query query = session.createQuery(hql);
4. //MyUser类的2个属性必须和:username和:password对应
5. MyUser myUser = **new** MyUser("name1","password1");
6. query.setProperties(myUser);
7. List<Users> list = query.list();
8. **for**(Users users : list){
9. System.out.println(users.getId());
10. }