

hibernate -- HQL语句总结 - 飘来荡去。 - 博客园(1)

[hibernate -- HQL语句总结](#)

1. 查询整个映射对象所有字段

```
//直接from  
m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  
String hql  
= "from  
Users";  
Query qu  
ery = sess  
ion.create  
Query(hq  
l);  
List users  
= query.li  
st();  
for(Users  
user : user  
s){  
System.o  
ut.println  
(user.get  
Name() +  
": "+ user  
.getPassw  
d() + ": "  
+ user.get  
Id());  
}  
输出结果  
为:  
name1 : p  
assword1  
: 1  
name2 : p  
assword2  
: 2  
name3 : p  
assword3  
: 3
```

2. 查询字段

```
//查询其中几个字段
String hql = "select name,password 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
List list = query.list();
for(Object[] object : list){
    String name = (String)object[0];
    String password = (String)object[1];
    System.out.println(name + " : " + password);
}
输出结果为:
name1 : password1
name2 : password2
name3 : password3
```

3.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

```

//查询其中几个字段,添加new list(),注意list里的是小写的。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,password) 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 = query.list();
for(List user : list){
String name = (String)user.get(0);
String password = (String)user.get(1);
System.out.println(name + " : " + password);
}
/**
输出结果为:
name1 : password1
name2 : password2
name3 : password3
*/

```

4.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

```

//查询其中几个字段,添加new map(),注意map里的是小写的。也不需要

```

这个而导入包，
这样通过
query.list
()出来的l
ist里存放
的不再是
默认的O
bject数组
了，而是
map集合
了
String hql
=" select
new map(
name,pas
swd) fro
m Users";
Query qu
ery = sess
ion.create
Query(hq
l);
//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
List list =
query.list
();
for(Map
user : list)
{
//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
//如果将h
ql改为：
String hql
=" select
new map(
name as u
sename,p
asswd as
password
) from Us
ers";那么
key将不
是字符串
0,1,2...了
，而是"u
sename",
"passwor
d"了
String na
me = (Stri
ng)user.ge
t("0");//ge
t("0");是g
et(key),
注意:0,1,
2...是字
符串，而
不是整形
String pas
swd = (St
ring)user.
get("1");
System.o

```

System.out.println
(name + "
: "+ pass
wd);
}
/**
输出结果
为:
name1 : p
assword1
name2 : p
assword2
name3 : p
assword3
*/

```

5.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

6.条件查询

```

//条件查
询，参数
索引值从
0开始，
索引位置
。通过se
tStringset
Parameter
设置参数
String hql
="from
Users wh
ere name=
? and pas
swd=?";
Query qu
ery = sess
ion.create
Query(hq
l);
//第1种方
式
// query.s
etString(0
,"name1"
);
// query.s
etString(1
,"passwo
rd1");
//第2种方
式
query.set
Parameter
(0,"name
1",Hibern
ate.STRI
NG);
query.set
Parameter
(1,"pass
word1",H
ibernate.S
TRING);
List list =
query.list
();
for(Users
users : list
){
System.o
ut.println
(users.get
Id());
}

```

```
//条件查询, 自定义索引名(参数名):
username,
password.通过set
String.set
Parameter
设置参数
String hql
="from
Users wh
ere name=
:username
and pass
wd=:pass
word";
Query qu
ery = sess
ion.create
Query(hq
l);
//第1种方
式
// query.s
etString("
username
", "name1
");
// query.s
etString("
password
", "passw
ord1");
//第2种方
式,第3个
参数确定
类型
query.set
Parameter
("usernam
e", "name
1",Hibern
ate.STRI
NG);
query.set
Parameter
("passwo
rd", "pass
word1",H
ibernate.S
TRING);
List list =
query.list
();
for(Users
users : list
){
System.o
ut.println
(users.get
Id());
}
```

```
//条件查询,通过setProperties设置参数
String hql = "from Users where name=:username and password=:password";
Query query = session.createQuery(hql);
//MyUser类的2个属性必须和:username和:password对应
MyUser myUser = new MyUser("name1","password1");
query.setProperties(myUser);
List list = query.list();
for(Users users : list){
    System.out.println(users.getId());
}
```

7.update 数据

执行SQL语句（为什么要用SQL语句，我想是为了执行某些复杂的SQL语句吧）

```
String sql = "update Table set field = 'test'";
Session session = HibernateSessionFactory.getSession();
session.createQuery(sql).executeUpdate();
ts.commit();
```

执行HQL语句

```
String hql = "update Table set field = 'test'";
```

```
Session session = HibernateSessionFactory.getSession();
Transaction ts = session.beginTransaction();
Query query = session.createQuery(hql);
query.executeUpdate();
ts.commit();
```