

# hibernate的hql查询语句总结 - xiaoluo501395377 - 博客园

这篇随笔将会记录hql的常用的查询语句，为日后查看提供便利。

在这里通过定义了三个类，Special、Classroom、Student来做测试，Special与Classroom是一对多，Classroom与Student是一对多的关系，这里仅仅贴出这三个bean的属性代码：

Special类：



复制代码

```
public class Special {
    private int id;
    private String name;
    private String type;
    private Set rooms;
    .....
}
```



复制代码

Classroom类：



复制代码

```
public class Classroom {
    private int id;
    private String name;
    private Special special;
    private Set students;
    .....
}
```



复制代码

Student类：



复制代码

```
public class Student {
    private int id;
    private String name;
    private String sex;
    private Classroom room;
    .....
}
```



复制代码

## 1. 最简单的查询

```
List specials = (List)session.createQuery("select spe from Special spe").list();
```

这是hql最基本的查询语句了，作用就是查出所有的Special对象放到一个List当中

## 2. 基于 ? 的参数化形式



复制代码

```
/**      * 查询中使用?, 通过setParameter的方式可以防止sql注入      * jdbc的setParameter的下标从1开始, hql的下标从0开始      */
List students = (List)session.createQuery("select stu from Student stu where name like ?")
    .setParameter(0, "%刘%")
    .list();
```



复制代码

在hql中同样支持基于 ? 的参数化形式查询，注意：在jdbc中，setParameter的下标是从1开始的，而hibernate的setParameter的下标是从0开始的。

## 3. 基于 :xx 的别名的方式设置参数

```
/**      * 在hql中可以使用别名的方式来查询, 格式是 :xxx 通过setParameter来设置别名      */
List students = (List)session.createQuery("select stu from Student stu where name like :name and sex like :sex")
    .setParameter("name", "%王%").setParameter("sex", "%男%")
    .list();
```

## 4. 如果返回的值只有一个，可以使用uniqueResult方法



复制代码

```
/**      * 如果得到的值只有一个，则可以使用uniqueResult方法      */
(Long)session.createQuery("select count(*) from Student stu where name like :name and sex like :sex")
    .setParameter("name", "%王%").setParameter("sex", "%男%")
    .uniqueResult();
/**      * 如果得到的值只有一个，则可以使用uniqueResult方法      */
Student stu = (Student)session.createQuery("select stu from Student stu where id = ?")
    .setParameter(0, 1)
    .uniqueResult();
```



复制代码

## 5. 基于投影的查询



复制代码

```
/**      * 基于投影的查询，如果返回多个值，这些值都是保存在一个object[]数组当中      */
List stus = (List)session.createQuery("select stu.name, stu.sex from Student stu where name like
                                     :name and sex like :sex")
.setParameter("name", "%张%").setParameter("sex", "%男%").list();
```



复制代码

## 6. 基于导航对象的查询



复制代码

```
/**      * 如果对象中有导航对象，可以直接通过对象导航查询      */      List stus =
(List)session.createQuery("select stu from Student stu where stu.room.name like :room and sex like :sex")
.setParameter("room", "%计算机应用%").setParameter("sex", "%女%")
.list();
```



复制代码

**注意:**若直接通过导航对象来查询时，其实际是使用cross join(笛卡儿积)来进行连接查询，这样做性能很差，不建议使用

## 7. 使用 in 进行列表查询



复制代码

```
/**      * 可以使用in设置基于列表的查询，使用in查询时需要使用别名来进行参数设置，      * 通
过setParameterList方法即可设置，在使用别名和?的hql语句查询时，?形式的查询必须放在别名前面      */
List stus = (List)session.createQuery("select stu from Student stu where sex like ? and stu.room.id in (:room)")//
.setParameter(0, "%女%").setParameterList("room", new Integer[]{1, 2})//
.list();      List stus = (List)session.createQuery("select stu from Student stu where stu.room.id in (:room) and
stu.sex like :sex")
.setParameterList("room", new Integer[]{1, 2}).setParameter("sex", "%女%").list();
```



复制代码

在使用 in 进行列表查询时，这个时候要通过 setParameterList() 方法来设置我们的参数，**注意：**如果一个参数通过别名来传入，一个是通过 ? 的方式来传入的话，那么通过别名的hql语句以及参数设置语句要放在 ? 的后面，不然hibernate会报错。如果都是使用 别名 来设置参数，则无先后顺序

## 8. 分页查询



复制代码

```
/**      * 通过setFirstResult(0).setMaxResults(10)可以设置分页查询，相当于offset和pagesize
*/      List stus = (List)session.createQuery("select stu from Student stu where stu.room.name like :room and sex
like :sex")
.setParameter("room", "%计算机应用%").setParameter("sex", "%女%").setFirstResult(0).setMaxResults(10)
.list();
```



复制代码

## 9. 内连接查询

```
/**      *      使用对象的导航查询可以完成连接查询，但是使用的是Cross Join(笛卡儿积)，效率不高，所以
建议使用join来查询      */      List stus = (List)session.createQuery("select stu from Student stu join
stu.room room where room.id=2")
.list();
```

在hql中使用连接查询的语句与我们的sql进行连接查询的语句是有区别的：

**hql:** select stu from Student stu join stu.room rooms **sql:** select t.\* from Student t join Classroom c on t.cid=c.id

## 10. 左外连和右外连查询

```
/**      *      左外连和右外连其实是相对的，left join 就是以左边的表为基准，right join 就是以右边的
表为基准      */      List stus = (List)session.createQuery("select room.name, count(stu.room.id) from
Student stu right join stu.room room group by room.id")
.list();
```

## 11. 创建DTO类，将查询出来的多个字段可以存放到DTO对象中去



复制代码

```
/**      * 如果我们查询出多个字段的话，通常会创建一个DTO对象，用来存储我们查询出来的数据，通过 new XXX() 这样的方式      * 前提是XXX这个类里面必须要有接受这些字段的构造方法才行，而且必须要使用类的全名      */
List stus = (List)session.createQuery("select stu.id,stu.name,stu.sex,room.name,special.name from Student stu left join stu.room room left join room.special special")//
        .list();//        for(Object[] obj : stus){
//        System.out.println(obj[0] + ", " + obj[1] + ", " + obj[2] + ", " + obj[3] + ", " + obj[4]);
}
```

```
List stus = (List)session.createQuery("select new com.xiaoluo.bean.StudentDTO(stu.id, stu.name, stu.sex, room.name, special.name) from Student stu left join stu.room room left join room.special special")
        .list();
```



复制代码

## 12. group having字句

```
/**      * 在hql中不能通过给查询出来的字段设置别名，别名只能设置在from 后面      */
List stus = (List)session.createQuery("select special.name, count(stu.room.special.id) from Student stu right join stu.room.special special group by special.id having count(stu.room.special.id)>150")
        .list(); // 查询出人数大于150个人的专业
```

```
// 查询出每个专业中男生与女生的个数
List stus = (List)session.createQuery("select special.name, stu.sex, count(stu.room.special.id) from Student stu right join stu.room.special special group by special.id,stu.sex")
        .list();
```

基本上用到的hql查询语句就是这些，以后若再遇到会进行补充。