1. HQL 基础
2. 定义

 HQL是面向对象的查询语言（Hibernate框架会将HQL语言解析成SQL然后执行数据库的查询）

1. 语句形式



(1) select子句：选择查询持久化对象中的指定属性，并指定数据类型返回

(2) from子句：用来指定hql语句的查询目标，即映射配置的持久化类及其属性

(3) where子句：逻辑表达式，用来设置查询条件，限制返回结果和范围

(4) group by子句：分组查询语句

(5) having子句：对分组进行限制条件设置

(6) order by子句：用来指定查询结果中的实例对象的排序

(7) From子句在HQL语句中不可或缺的组成部分，一个最简单的HQL语句形式只要有from就可以了，其他的子句都可以省略，这点与SQL语句不同

1. 区别
2. HQL是面向对象的查询语言，对Java类与属性大小写敏感
3. HQL对关键字不区分大小写（习惯小写）
4. HQL查询步骤
5. org.hibernate.Query接口：定义有执行查询的方法，该方法完成HQL语句的解析和执行过程，并返回查询结果；
6. 创建Query实例：①Session的createQuery()方法创建Query实例；②createQuery方法包含一个HQL语句参数，createQuery(hql)
7. 执行查询：Query接口的list()方法执行HQL查询，list()方法返回结果数据类型为java.util.List，List集合中存放符合查询条件的持久化对象。
8. 实例代码：

String sql = "from Hib";

Query query = session.createQuery(sql);

List<Hib> hList = query.list();

for(Hib h : hList){

System.out.println(h);

}

1. HQL语句详解
2. from检索语句
3. 不需要引入持久化类的全限定名（如from com.imooc.model.Seller），直接引入类名

(from Seller)即可

(2) HQL语句中可以省略全限定名，是因为auto-import自动引入缺省情况，hibernate框架中，在解析HQL语句时，会根据映射配置信息自动完成持久化类的导入。

(3) 在from子句中全限定名的引用与直接引入类名是一样，并不会报错！

(4) 为持久化类指定别名：from Seller as s/from Seller s

1. select选择语句
   1. 返回Object[] 结果
2. select子句中未指定返回数据类型，默认为Object[]。HQL的select查询语句中，如果指定了多个查询字段，则返回的是一个Object[]数组；如果只指定了一个查询字段，则返回的是一个Object对象。
3. 实例代码：

String hql="select s.name,s.tel,s.address from Seller s";

Query query=session.createQuery(hql);

List<Object[]> lists=query.list();

for (Object[] list : lists) {

System.out.println("name:"+list[0]);

System.out.println("tel:"+list[1]);

System.out.println("address"+list[2]);

}

* 1. 返回List查询结果

1. 实例代码：

String hql = "select new list(s.name,s.tel,s.address) from Seller s";

...

List<List>lists=query.list();

2.3 返回map形式查询结果

(1) 以new map(s.name,s.tel,s.address) from 持久化的实体类；

(2) key值为索引值，为字符串类型,"0","1"等；

(3) 若在map中使用了别名：(s.name as n,s.tel as t,s.address as a) 就可以使用map.get(别名)。 这是as关键字是必须的。

(4) 实例代码：

String hql = "select new map(s.id as id,s.name as name,s.tel as tel) from Seller as s";

Query query = session.createQuery(hql);

List<Map> maps = query.list();

for (Map map : maps) {

System.out.println(map.get("id")+" "+map.get("name")+" "+map.get("tel"));

}

2.4 以自定义类型返回查询结果

(1) 先创建自定义构造器

public Sellers(String name,String address){

this.name=name;

this.address=address;

}

1. 在 hql中 select new Seller（s.name,s.address） from Seller s
2. 创建List集合

List<Seller> sellers=query.list();

for(Sellers selle:sellers){

输出自己想要的字段（构造器中定义的）

}

2.5 持久化类无参构造器

(1) 增加自定义构造器，一定要补充默认构造器，否则hql=" from classname" 会出错，因为，在Hibernate没有指定的查询的放回集合时候，Hibernate会自动去找默认构造器

2.6 distinct关键字

(1) String hql="select distinct c.sex from Customer c";

(2) distinct使用时只能放在查询语句的第一个字段前，且只对查询单个字段才有过滤效果。

(3) 如果查询语句中包含多个字段，那么distinct将不再具有过滤功能，但运行也不会报错。

(4) 如果distinct没有放在查询语句的第一个字段前, 或语句存在多个distinct，运行会出错。

3. where限制语句

3.1 比较运算

(1) = <> < > >= <=

(2) null值判断 -- x = null →x is null；x <> null → x is not null

* 1. 范围运算：in 和 between

1. String hql = "from Customer as c where c.age in(20,40)";
2. String hql = "from Customer as c where c.age not in(20,40)";
3. String hql = "from Customer as c where c.age between 20 and 40";
4. String hql = "from Customer as c where c.age not between 20 and 40";
   1. 模糊查询匹配
5. 实例代码：

String hql = "from Customer as c where c.name like '张\_'";

String hql = "from Customer as c where c.address like '%北京%'";

3.4 逻辑运算

(1) and（逻辑与） or（逻辑或） not（逻辑非）

(2) 实例代码：

String hql="from Commodity c where c.price between 100 and 4000 and c.category like '%电 脑%'";

String hql1="from Commodity c where c.price between 100 and 4000 or c.category like '%电 脑%'";

* 1. 集合运算(一对多，实体类中含有一个集合)

String hql = "from Order where o.orderItems is not empty ";

3.5 四则运算

(1) HQL语句中也可以使用 + - \* / 四则运算

(2) 四则运算可以在where子句和select子句中使用

* 1. 查询单个对象 – 而不是List对象

1. 实例代码：

String hql = "from Customer as c where c.id=1";

Query query = session.createQuery(hql);

Customer c = (Customer)query.uniqueResult();

System.out.println(c);

1. 要保证查询结果只有一个或者没有，不然会抛出异常。

4. 排序 – order by

(1) 升序排序 asc(默认) 降序排序 desc

(2) 实例代码：

String hql = "from Commodity order by price asc"; // 升序

String hql = "from Commodity order by price desc"; // 降序

String hql = "from Commodity order by seller.id asc ,price desc, name asc";//设置多个排序规则用逗号隔开