

HQL: hibernate query language: hibernate的查询语言。

opaque=true设置文本域的颜色为不透明。

语法和SQL语法非常类似。

1, 单表查询

(1) 基本全表查询:

Query query = session.createQuery("from person");//返回一个查询的结果集

```
List <person>list = query.list();
for(person person:list){
    System.out.println(person.getName());
}
```

(2) 别名查询:

Query query = session.createQuery("from person p");//返回一个查询的结果集, p为person的别名

```
List <person>list = query.list();
for(person person:list){
    System.out.println(person.getName());
}
```

别名查询一般要在多表查询中使用

(3) 条件查询:

hibernate5.4.x查询使用jpa style:

```
Query query = session.createQuery("from person where name= ?0 ");
query.setParameter(0, "21a12nqili");
```

hibernate5.4.x以前版本查询使用setParameter:

```
Query query = session.createQuery("from person where name=? ");
query.setParameter(0, "21a12nqili");
```

(4) 具名查询:

Query query = session.createQuery("from person where name=:为name字段重新起的名字");

```
query.setParameter("为name字段重新起的名字", "21a12nqili");
List <person>list = query.list();
```

(5) 分页查询: (使用的是一个方法, 而不是一个关键字)

```
Query query = session.createQuery("from person");
```

```
//设置起始行，默认从0开始
query.setFirstResult(2);
//设置查询的行数
query.setMaxResults(3);
List<person>list = query.list();
```

(6) 查询排序:

```
Query query = session.createQuery("from person order by id desc");
List<person> list = query.list();
```

其中desc表示降序排序，asc表示升序。

(7) 聚合查询:

```
Query query = session.createQuery("select count(*) from person order by id desc");
```

```
List<Long> list = query.list();
System.out.println(list.get(0));
```

或

```
Long countLong =(Long)query.uniqueResult();
System.out.println(countLong);
```

(8) 投影查询: (就是查询局部的字段)

方案一: 不对对象进行封装的时候:

```
Query query = session.createQuery("select id,name from person");
//封装称为一个对象数组
List<Object[]> list = query.list();
for(Object object []:list){
    for(Object object2:object){
        System.out.print(object2+" ");
    }
    System.out.println();
}
```

方案二: 将对象进行封装

```
Query query = session.createQuery("select new person(id,name) from person");
```

```
//封装称为一个对象
```

//需要在实体类中写一个含有参数的构造方法，当写实体类的时候，既要有一个无参的构造函数，也要有一个有参的构造函数

```
//构造函数的参数要根据需要封装的实体的参数进行编写
```

```
List<person> list = query.list();
```

```
for(person p1:list){  
    System.out.println(p1);  
}
```

2, 多表查询

hibernate提供的多表查询

1) 内连接查询: inner join

```
Query query = session.createQuery("select p1.id,p1.name from t_person_fk p1  
inner join t_person_fk p on p1.name=?0");  
    query.setParameter(0, "2213");  
    List<person> list = query.list();  
    System.out.println(list.toString());
```

2) 左连接查询: left join

效果: 左边的数据会全部显示, 右边的数据如果数据库中没有数据, 则显示为null

3) 右连接查询: right join

效果: 右边的数据会全部显示, 左边的数据如果数据库中没有数据, 则显示为null