JPA-jpql查询常用方法

1 统计查询

- jpql不允许写count(*)

```
@Query(" select count(t) from FollowerInfo t where investUserId = :invUserId")
Integer findFollowerNumberByInvUserId(@Param("invUserId") Long invUserId);
```

2 嵌套糊模查询

```
@Query("select p from PrpDcustomer p where customerCode in (select customerCode from PrpDcustomer where customerCName
like :customerName%)”)
public List<PrpDcustomer> queryJpql(@Param(“customerName”) String customerName);//按名称参数注入
```

```
@Query(value = "select p from PrpDcustomer p where customerCode in(select u.userCode from PrpDjuser u)",nativeQuery = true)
```

3 聚合查询

- dao层：

```
@Query(value = "select new com.spring.boot.entity.CountEntity(count(p),max(p.customerCode)) from PrpDcustomer p group
by p.customerType")
public List<CountEntity> queryJpql();
```

- 实体类：
  以上查询返回的对象是自定的对象 ，因为聚合后的值都是原对象不存在的。
  Sql语句里的字段顺序要与对象构造函数一致，且count只能用long类型。

```
public class CountEntity implements Serializable {
private  Long m;
private  String  n;
public CountEntity(){}
public CountEntity(Long m, String n) {
this.m = m;
this.n = n;
}
//省略 get set
}
```

4 修改字段

- 修改必须加上@Modifying(clearAutomatically = true)

```
@Transactional
@Modifying(clearAutomatically = true)//修改必必须加上
@Query(value = "update UserInfo  set isEnabled = abs(1-isEnabled) where id=?1")
public int updateEnabled(int id);//按位参数注入
```

5 Specifications分页条件查询

参考

```
com.sinosoft.demo.core.customer.service.impl;
public PageInfo<PrpDcustomerDto> queryPaging(PrpDcustomerQueryConditionDto queryDto);
```

6 jpql分页参数查询

6.1 List返回:

`PrpDcustomerServiceImpl.java`

```
Pageable pageable=new PageRequest(0,3,null);
return prpDcustomerDao.queryJpql(pageable);
```

`PrpDcustomerDao.java`

```
@Query(value = "select p from PrpDcustomer p ")
public List<PrpDcustomer> queryJpql(Pageable pageable);//Pageable参数只能放在最后一个
```

> 注意：Pageable参数只能放在最后一个

6.2 Page分页对象返回:

`PrpDcustomerServiceImpl.java`

```
public Page<PrpDcustomer> queryJpql() {
Pageable pageable=new PageRequest(0,3,null);
return prpDcustomerDao.queryJpql(pageable);//Pageable参数只能放在最后一个
}
```

`PrpDcustomerDao.java`

```
@Query(value = "select p from PrpDcustomer p ")
public Page<PrpDcustomer> queryJpql(Pageable pageable);
```

7 Jpql对象传参

7.1 按名称入参

```
@Query("select p from PrpDcustomer p where customerCName like :customerName%)” )
public List<PrpDcustomer> queryJpql(@Param(“customerName”) String customerName);//按名称参数注入
```

7.2 按位入参

```
@Query(value = "update UserInfo  set isEnabled = abs(1-isEnabled) where id=?1")
public int updateEnabled(int id);//按位参数注入
```

7.3 按对象入参

```
@Query(value = "select p from PrpDcustomer p where p.customerCode=:#{#pc.customerCode} and p.customerCName=:#{#pc.customerCName}")
public Page<PrpDcustomer> queryJpql(@Param("pc") PrpDcustomer pc, Pageable pageable);
```

8 原生sql查询

```
@Query(value = "select * from PrpDcustomer where CUSTOMERCNAME LIKE :name%",nativeQuery = true)
public Page<PrpDcustomer> queryJpql(@Param("name") String name);
```

9 EntityManager自定义查询

```
String dataSql="select * from PrpDcustomer where customerCName like :customerCName";
Query dataQuery=entityManager.createNativeQuery(dataSql,PrpDcustomer.class);
dataQuery.setParameter("customerCName","张%");

List<PrpDcustomer> list=dataQuery.getResultList();
```

10 查询自定义字段

sql:

```
String dataSql="select new PrpDuser(p.userCode as userCode,p.createdBy as createdBy,p.createdTime as createdTime,p.sex as sex,p.userName as userName) from PrpDuser p”；
```

```
String dataSql="select new PrpDuser(substring(p.userCode,1,1),p.createdBy,p.createdTime,p.sex,p.userName) from PrpDuser p";
```

实体类-自定义字段实例对象时一定要声名一个对应的构造函和默认构造函数 ：

```
public PrpDuser(){}
public PrpDuser(String userCode, String createdBy, Date createdTime, String sex, String userName) {
this.userCode=userCode;
this.createdBy=createdBy;
this.createdTime=createdTime;
this.sex=sex;
this.userName=userName;
}
```