class Entity1{

} QEntity1

class Entity2{

private Entity1 prop;

} QEntity2

selectFrom（QEntity2.ins)

.leftjoin(QEntity1.ins)

.on（ QEntity2.ins.id= 1.id）

selelctFrom（QEntity2.ins）

.leftJoin（QEntity2.ins.prop）

.fetchOne() 1 JPQL -- sql （select subselect)

select \* from Entity2; （1）

Select × from entity1; （100）

.fetch（) ---

.fetchJoin()

select × from Entity2 left join Entity1（1）

select a.\*， b, \* from Entity1 a fetch left join Entity2 b on a. = b.

JAP delete（id）

1. 查出所有的Id
2. 根据Id进行删除

对App的接口

1. mybatis <delete or udpate> 原生SQL
2. QueryDsl Delete update的写法

Entity1 Entity2 1：2

1. 根据子查询的方式查出Id

wms(listByCondition<List>

count fech().count（）；

1. 再去根据Id来查询出所有的对象。

查询一个自定义类出来。

Entity(非常复杂）

强类型（javascript=》 typescript）

fetch（ class, p)

Map