

```

import os,django
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "Jango_lesson_157.settings")
django.setup()

from relationapp.models import Person,Passport,Category,Goods,Student,Course
from relationapp import models
# 一对一的查询

# 1、只查询一方的数据:查单表
# 只查询人的信息
...

pers = Person.objects.all()
print(pers)
per = Person.objects.filter(name="Tom")
print(per)

# 只查护照的信息
pas = Passport.objects.all()
print(pas)

...

# 2、通过一方，再查另一方：需要先获取一方的数据
# 通过人查护照
...

per = Person.objects.filter(name="Tom")[0]
print(per)

print(per.passport)    #没有属性时 对象名.对方类名小写

# 通过护照来查人
port = Passport.objects.get(country="英国")
print(port.per)    # 在passport中有per属性
...

# 3、通过一方查另一方（查询条件是对方的列）
# 查护照为中国的人的信息
...

per = Person.objects.filter(passport__country="中国")
print(per)

pers = Person.objects.filter(passport__country__contains="国")
print(pers)

# 查询名字为Tom的护照的信息
ports = Passport.objects.filter(per__name="Jack")
print(ports)

```

```

print(ports[0].country)
'''

# 4、保留双方的数据
# 查询名字为Tom的护照的信息及人的信息
'''

ports =
Passport.objects.filter(per__name="Tom").values("country", "expire", "per__name", "per__age")
print(ports)

pers =
Person.objects.filter(name="Tom").values("name", "age", "passport__country", "passport__expire")
print(pers)
'''

# 查询护照为法国的人的信息和护照的信息
# pers = Person.objects.filter(passport__country="法
国").values("name", "age", "passport__country", "passport__expire")
# print(pers)
#
# ports = Passport.objects.filter(country="法
国").values("country", "expire", "per__name", "per__age")
# print(ports)

# 一对多查询 分类 和 商品
# 1、只查询一方
'''

# 只查询分类
cats = Category.objects.all()
print(cats)

#只查询商品的信息
goods = Goods.objects.all()
print(goods)
'''

# 2、通过一方，查询另一方（先获取一方的数据）
# 先查询一个商品，再查该商品所属的分类
'''

goods = Goods.objects.get(pk=1)
print(goods)
print(goods.cate)

# 先查一个分类，再查该分类下的商品的信息
cate = Category.objects.get(title="家电")
print(cate.goods_set)
print(type(cate.goods_set)) # 返回值是一个RelatedManager
print(cate.goods_set.all())
'''

```

```

# 3、通过一方查询另一方（ 查询条件是对方的列 ）
# 查询商品名为男装的分类的信息
...

cate = Category.objects.filter(goods__title="男装").values()
print(cate)

#查询分类为家具的商品信息
goods = Goods.objects.filter(cate__title="家具")
print(goods)
...

# 4、保留双方数据

#查询分类为家具的商品信息和分类信息

# goods = Goods.objects.filter(cate__title="家具").values("title","price","cate__title","cate__note")
# print(goods)
#
# cates = Category.objects.filter(title="家具").values("title","note","goods__title","goods__price")
# print(cates)

#查询商品名为男装的分类的信息和商品

# cates = Category.objects.filter(goods__title="男装").values("title","note","goods__title","goods__price")
# print(cates)

# goods = Goods.objects.filter(title="男装").values("title","price","cate__title","cate__note")
# print(goods)

#查询商品价格大于200的分类信息
# cate = Category.objects.filter(goods__price__gt=200)
# print(set(cate))

# 多对多查询 学生和课程

...

#1、只查询一方的数据
# 只查询学生的信息
stus = Student.objects.all()
print(stus)

#只查询课程的信息
course = Course.objects.all()
print(course)

```

```
#2、通过一方查询另一方（先获取一方的数据）
# 先查询课程，再根据课程查学生
c = Course.objects.get(title="HTML")
print(c.stu) # RelatedManager
print(c.stu.all())

# 查询Tom 所选的课程
s = Student.objects.get(name="Tom")
print(s.course_set)
print(s.course_set.all().values())

...

# 3、通过一方查询另一方（查询条件是对方的列）
# 查询课程为MySQL的学生的信息
...

stus = Student.objects.filter(course__title="MySQL")
print(stus)

# 查询学生为Jack的课程信息
cs = Course.objects.filter(stu__name="Jack")
print(cs)

...

# # 查询课程为MySQL的学生的信息
# c = Course.objects.get(title="MySQL")
# c.stu.all()

# 4、保留双方的数据

#查询学生为Jack的课程信息和学生信息
# c = Course.objects.filter(stu__name="Jack").values("title","duration","stu__name","stu__age")
# print(c)
```