# Django ORM查询

queryset是查询集，Django会对查询返回的结果集QerySet进行缓存，这里是为了提高查询效率，也就是说，在你创建一个QuerySet对象的时候，Django并不会立即向数据库发出查询命令，只有在你需要用到这个QuerySet的时候才回去数据库查询

返回QuerySet对象的方法

all() #返回表中所有数据  
filter() #返回符合条件的数据  
exclude() #返回不符合条件的数据  
order\_by() #返回查询结果集进行排序  
reverse() #对排序的结果反转  
distinct() #返回去重条件的数据

特殊的QuerySet

values() 返回一个可迭代的字典序列  
values\_list() 返回一个可迭代的元祖序列

返回具体对象

get() #返回满足条件的对象  
first() #返回第一条数据  
last() #返回最后一条数据

返回布尔值

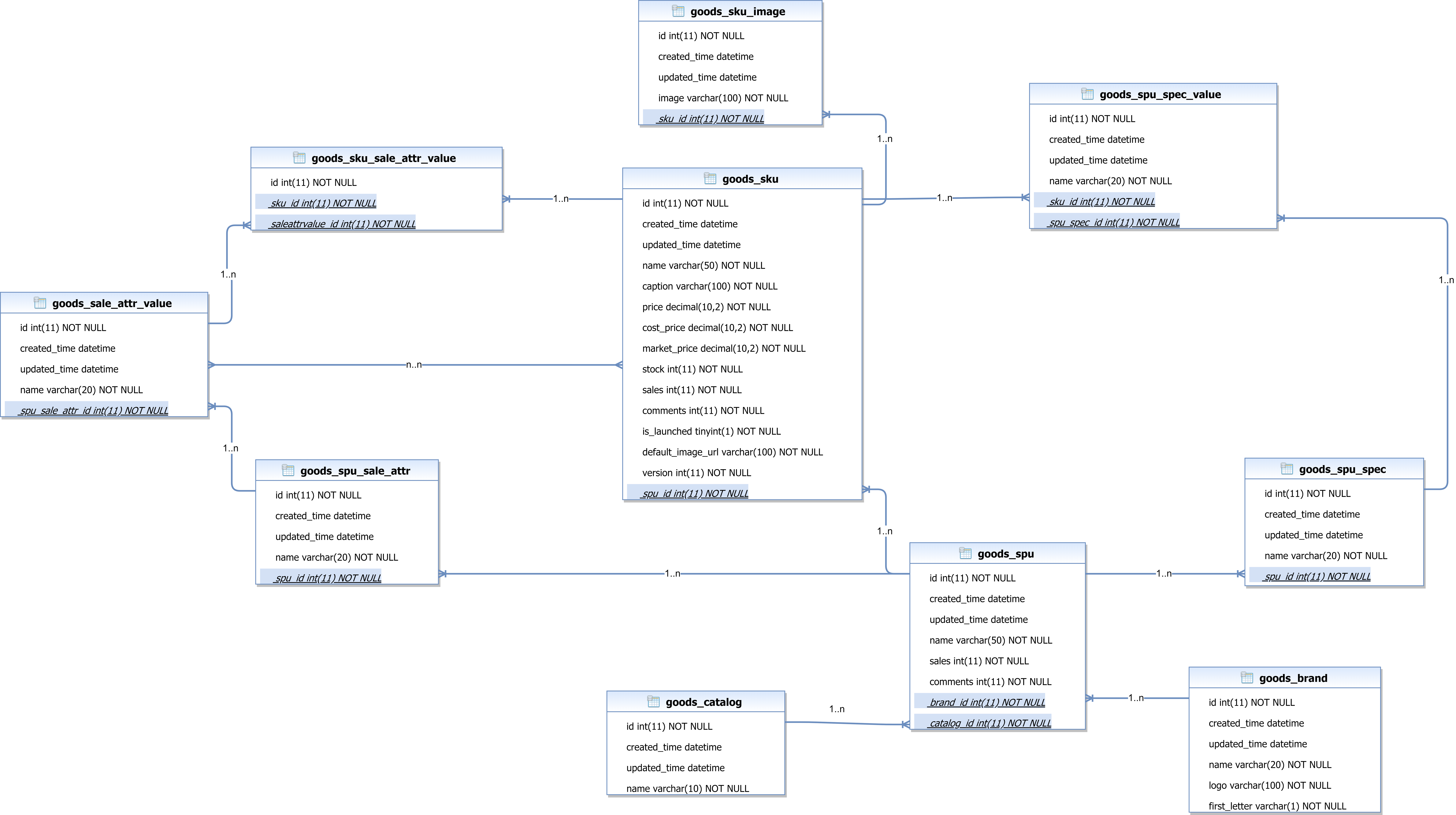
exists() #判断查询的数据是否存在

返回数字

count() #返回查询集中对象的数目

## dadashop商品部分ER图

**后面会用部分表**



查看所执行的SQL语句，query.\_\_str\_\_() , 或者 print(QuerySet.query)

例子：  
In： SKU.objects.filter(id=1).query.\_\_str\_\_()  
  
结果：  
<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>]>  
  
Out: 'SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` WHERE `DDSC\_SKU`.`id` = 1'

MySQL开启日志记录查询

SHOW VARIABLES LIKE "general\_log%";  
SET GLOBAL general\_log = 'ON';  
SET GLOBAL general\_log\_file = '/var/log/mysql/general\_log.log';

## 基础查询

#### 查询所有对象

all() 方法

查询sku表中所有数据

例子：   
In: SKU.objects.all()  
  
结果：   
<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>, <SKU: 2: 安踏A红色大尺寸>, <SKU: 3: 安踏B红色大尺寸>]>  
  
  
In: SKU.objects.all().query.\_\_str\_\_()  
  
out：'SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU`'

#### 查询单个对象

get 如果没有查询匹配的结果,get()则会引发DoseNotExist异常,有多个结果也会返回MultipleObjectsReturned异常

查询sku表id为1的数据

例子:  
In：SKU.objects.get(id=1)  
  
结果：  
<SKU: 1: 安踏A蓝色小尺寸>  
  
out: <SKU: 1: 安踏A蓝色小尺寸>

DoseNotExist 异常展示

例子：  
SKU.objects.get(sales=200)  
  
goods.models.DoesNotExist: SKU matching query does not exist.

MultipleObjectsReturned 异常展示

例子：  
SKU.objects.get(cost\_price=1000)  
  
goods.models.MultipleObjectsReturned: get() returned more than one SKU -- it returned 3!

first 第一个值

例子：  
SKU.objects.first()  
  
结果：  
<SKU: 1: 安踏A蓝色小尺寸>  
  
SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` ORDER BY `DDSC\_SKU`.`id` ASC LIMIT 1

last 最后一个值

例子：  
SKU.objects.last()  
  
结果：  
<SKU: 3: 安踏B红色大尺寸>  
  
SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` ORDER BY `DDSC\_SKU`.`id` DESC LIMIT 1

#### 字段查找

##### 比较匹配：

大于\_\_gt

例子：  
In: SKU.objects.filter(id\_\_gt="1").query.\_\_str\_\_()  
  
结果：<QuerySet [<SKU: 2: 安踏A红色大尺寸>, <SKU: 3: 安踏B红色大尺寸>]>  
  
Out: 'SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` WHERE `DDSC\_SKU`.`id` > 1'

小于\_\_lt

例子：  
In: SKU.objects.filter(id\_\_lt="2").query.\_\_str\_\_()  
  
结果：<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>]>  
  
Out: 'SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` WHERE `DDSC\_SKU`.`id` < 2'

##### 范围内：

\_\_in = ["大","小",] 在这个范围内

在这个列表里边的值

例子：  
In: SKU.objects.filter(id\_\_in=["1","2"]).query.\_\_str\_\_()  
  
结果：  
<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>, <SKU: 2: 安踏A红色大尺寸>]>  
  
Out: 'SELECT `DDSC\_SKU`.`id`, `DDSC\_SKU`.`create\_time`, `DDSC\_SKU`.`update\_time`, `DDSC\_SKU`.`name`, `DDSC\_SKU`.`caption`, `DDSC\_SKU`.`SPU\_ID\_id`, `DDSC\_SKU`.`price`, `DDSC\_SKU`.`cost\_price`, `DDSC\_SKU`.`market\_price`, `DDSC\_SKU`.`stock`, `DDSC\_SKU`.`sales`, `DDSC\_SKU`.`comments`, `DDSC\_SKU`.`is\_launched`, `DDSC\_SKU`.`default\_image\_url`, `DDSC\_SKU`.`version` FROM `DDSC\_SKU` WHERE `DDSC\_SKU`.`id` IN (1, 2)'

## 正向查询

ForeignKey操作

对象查找（跨表）  
语法：  
对象.关联字段.字段

例子：通过sku表id为1的查询对应的spu表商品名

SKU.objects.first().spu.name  
  
结果：  
'安踏A'

## 反向查询

ForeignKey操作

对象查找（跨表）  
语法：  
obj.表名\_set

例子：通过spu表id为1查询对应的sku表商品名

spu\_obj = SPU.objects.get(id=1)  
spu\_obj.sku\_set.all()  
  
结果：  
<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>, <SKU: 2: 安踏A灰色大尺寸>]>

## 业务查询

### 查询商品id为1对应spu的总销量是多少

skus = SKU.objects.filter(id=1)  
for sku in skus:  
 print(sku.spu.sales)  
   
结果：  
0  
  
'SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE `goods\_sku`.`id` = 1'  
  
'SELECT `goods\_spu`.`id`, `goods\_spu`.`created\_time`, `goods\_spu`.`updated\_time`, `goods\_spu`.`name`, `goods\_spu`.`sales`, `goods\_spu`.`comments`, `goods\_spu`.`brand\_id`, `goods\_spu`.`catalog\_id` FROM `goods\_spu` WHERE `goods\_spu`.`id` = 1'

### 查询商品的所有品类

Catalog.objects.all()  
  
结果:  
<QuerySet [<Catalog: 手提包>]>  
  
'SELECT `goods\_catalog`.`id`, `goods\_catalog`.`created\_time`, `goods\_catalog`.`updated\_time`, `goods\_catalog`.`name` FROM `goods\_catalog` LIMIT 21'

### 查询手提包类别下有哪些SPU商品

catalog = Catalog.objects.get(name='手提包')  
catalog.spu\_set.all()  
  
结果：  
<QuerySet [<SPU: 安踏A>, <SPU: 安踏B>, <SPU: 安踏C>]>  
  
  
'SELECT `goods\_catalog`.`id`, `goods\_catalog`.`created\_time`, `goods\_catalog`.`updated\_time`, `goods\_catalog`.`name` FROM `goods\_catalog` WHERE `goods\_catalog`.`name` = '手提包''  
  
'SELECT `goods\_spu`.`id`, `goods\_spu`.`created\_time`, `goods\_spu`.`updated\_time`, `goods\_spu`.`name`, `goods\_spu`.`sales`, `goods\_spu`.`comments`, `goods\_spu`.`brand\_id`, `goods\_spu`.`catalog\_id` FROM `goods\_spu` WHERE `goods\_spu`.`catalog\_id` = 1 LIMIT 21'

### 查询类别为手提包对应在线的SKU商品

catalog = Catalog.objects.get(name='手提包')  
catalog.spu\_set.all().values('id')  
for spu\_id in spu\_ids:  
 sku = SKU.objects.filter(spu=spu\_id['id'],is\_launched=True)  
 print(sku)  
  
结果：  
<QuerySet [<SKU: 1: 安踏A蓝色小尺寸>, <SKU: 2: 安踏A灰色大尺寸>]>  
<QuerySet [<SKU: 3: 安踏B蓝色小尺寸>]>  
<QuerySet []>  
  
SELECT `goods\_catalog`.`id`, `goods\_catalog`.`created\_time`, `goods\_catalog`.`updated\_time`, `goods\_catalog`.`name` FROM `goods\_catalog` WHERE `goods\_catalog`.`name` = '手提包'  
  
SELECT `goods\_spu`.`id` FROM `goods\_spu` WHERE `goods\_spu`.`catalog\_id` = 1 LIMIT 21  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE (`goods\_sku`.`spu\_id` = 1 AND `goods\_sku`.`is\_launched` = 1) LIMIT 21  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE (`goods\_sku`.`spu\_id` = 2 AND `goods\_sku`.`is\_launched` = 1) LIMIT 21  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE (`goods\_sku`.`spu\_id` = 3 AND `goods\_sku`.`is\_launched` = 1) LIMIT 21

### 查询SKU商品id是1的详情图

SKUImage.objects.filter(sku=1)  
  
结果：  
<QuerySet []>  
  
SELECT `goods\_sku\_image`.`id`, `goods\_sku\_image`.`created\_time`, `goods\_sku\_image`.`updated\_time`, `goods\_sku\_image`.`sku\_id`, `goods\_sku\_image`.`image` FROM `goods\_sku\_image` WHERE `goods\_sku\_image`.`sku\_id` = 1 LIMIT 21  
  
  
  
  
SKU.objects.first().skuimage\_set.all()  
  
结果：  
<QuerySet []>  
  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` ORDER BY `goods\_sku`.`id` ASC LIMIT 1  
  
SELECT `goods\_sku\_image`.`id`, `goods\_sku\_image`.`created\_time`, `goods\_sku\_image`.`updated\_time`, `goods\_sku\_image`.`sku\_id`, `goods\_sku\_image`.`image` FROM `goods\_sku\_image` WHERE `goods\_sku\_image`.`sku\_id` = 1 LIMIT 21

### 查询SKU表最后一个商品所对应的销售属性名

spu\_id = SKU.objects.last().spu.id  
sales = SPUSaleAttr.objects.filter(spu=spu\_id)  
for sale in sales:  
 print(sale.name)  
   
结果：  
安踏B/尺寸  
安踏B/颜色  
  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` ORDER BY `goods\_sku`.`id` DESC LIMIT 1  
  
SELECT `goods\_spu`.`id`, `goods\_spu`.`created\_time`, `goods\_spu`.`updated\_time`, `goods\_spu`.`name`, `goods\_spu`.`sales`, `goods\_spu`.`comments`, `goods\_spu`.`brand\_id`, `goods\_spu`.`catalog\_id` FROM `goods\_spu` WHERE `goods\_spu`.`id` = 2  
  
SELECT `goods\_spu\_sale\_attr`.`id`, `goods\_spu\_sale\_attr`.`created\_time`, `goods\_spu\_sale\_attr`.`updated\_time`, `goods\_spu\_sale\_attr`.`spu\_id`, `goods\_spu\_sale\_attr`.`name` FROM `goods\_spu\_sale\_attr` WHERE `goods\_spu\_sale\_attr`.`spu\_id` = 2

查询SKU表第一个商品所对应的销售属性值

sales = SKU.objects.first().sale\_attr\_value.all()  
for sale in sales:  
 print(sale.name)  
   
   
结果:  
15寸  
蓝色  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` ORDER BY `goods\_sku`.`id` ASC LIMIT 1  
  
SELECT `goods\_sale\_attr\_value`.`id`, `goods\_sale\_attr\_value`.`created\_time`, `goods\_sale\_attr\_value`.`updated\_time`, `goods\_sale\_attr\_value`.`spu\_sale\_attr\_id`, `goods\_sale\_attr\_value`.`name` FROM `goods\_sale\_attr\_value` INNER JOIN `goods\_sku\_sale\_attr\_value` ON (`goods\_sale\_attr\_value`.`id` = `goods\_sku\_sale\_attr\_value`.`saleattrvalue\_id`) WHERE `goods\_sku\_sale\_attr\_value`.`sku\_id` = 1

### 查询SKU表商品为2对应的品牌名

SKU.objects.get(id=2).spu.brand.name  
  
结果：  
'安踏'  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE `goods\_sku`.`id` = 2  
  
SELECT `goods\_spu`.`id`, `goods\_spu`.`created\_time`, `goods\_spu`.`updated\_time`, `goods\_spu`.`name`, `goods\_spu`.`sales`, `goods\_spu`.`comments`, `goods\_spu`.`brand\_id`, `goods\_spu`.`catalog\_id` FROM `goods\_spu` WHERE `goods\_spu`.`id` = 1  
  
SELECT `goods\_brand`.`id`, `goods\_brand`.`created\_time`, `goods\_brand`.`updated\_time`, `goods\_brand`.`name`, `goods\_brand`.`logo`, `goods\_brand`.`first\_letter` FROM `goods\_brand` WHERE `goods\_brand`.`id` = 1

### 查询SKU在线商品销量排名

SKU.objects.filter(is\_launched=True).order\_by('sales')  
  
结果：  
<QuerySet [<SKU: 3: 安踏B蓝色小尺寸>, <SKU: 1: 安踏A蓝色小尺寸>, <SKU: 2: 安踏A灰色大尺寸>]>  
  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE `goods\_sku`.`is\_launched` = 1 ORDER BY `goods\_sku`.`sales` ASC LIMIT 21

### 查询SKU商品为1的销售规格名和销售规格值

spec\_values = SKU.objects.get(id=2).skuspecvalue\_set.all()  
for spec\_value in spec\_values:  
 print(spec\_value.spu\_spec.name)  
 print(spec\_value.name)  
  
  
  
SELECT `goods\_sku`.`id`, `goods\_sku`.`created\_time`, `goods\_sku`.`updated\_time`, `goods\_sku`.`name`, `goods\_sku`.`caption`, `goods\_sku`.`spu\_id`, `goods\_sku`.`price`, `goods\_sku`.`cost\_price`, `goods\_sku`.`market\_price`, `goods\_sku`.`stock`, `goods\_sku`.`sales`, `goods\_sku`.`comments`, `goods\_sku`.`is\_launched`, `goods\_sku`.`default\_image\_url`, `goods\_sku`.`version` FROM `goods\_sku` WHERE `goods\_sku`.`id` = 2  
  
SELECT `goods\_spu\_spec\_value`.`id`, `goods\_spu\_spec\_value`.`created\_time`, `goods\_spu\_spec\_value`.`updated\_time`, `goods\_spu\_spec\_value`.`sku\_id`, `goods\_spu\_spec\_value`.`spu\_spec\_id`, `goods\_spu\_spec\_value`.`name` FROM `goods\_spu\_spec\_value` WHERE `goods\_spu\_spec\_value`.`sku\_id` = 2