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

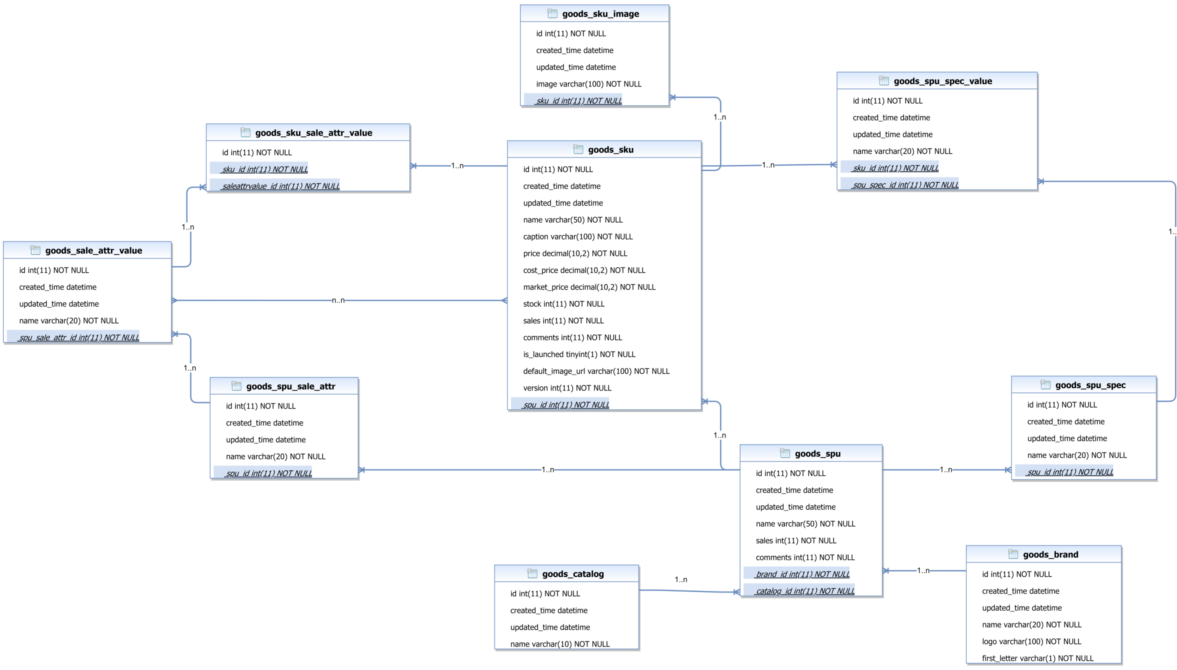
```
exists()   #判断查询的数据是否存在
```

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
```