

محمد هادی امید

تمرین ۱۵

سوال سوم:

در قبل که مدل های خود را برای فروشگاه پیاده کرده اید برای آن ابتدا چند محصول وارد کنید. سپس کوئری های زیر را پیاده سازی کنید

```
from django.contrib.auth.models import User
from order.models import *
from product.models import *
```

```
product1 = Product.objects.create(name='Samsung A71s', buy_price=899, sell_price=999, qty=20,
supplier=User.objects.create(username='Ahmad', password='1111'),
sub_category=SubCategory.objects.create(name='Sam Phone',
category=Category.objects.create(name='Samsung Co.')))
```

```
In [26]: product1 = Product.objects.create(name='Samsung A71s', buy_price=899, sell_price=99
...: 9, qty=20, supplier=User.objects.create(username='Ahmad', password='1111'), sub_cat
...: egory=SubCategory.objects.create(name='Sam Phone', category=Category.objects.create
...: (name='Samsung Co.')))
```

```
User2 = User.objects.get(id=2)
```

```
sub_category2 = SubCategory.objects.get(id=2)
```

```
product2 = Product.objects.create(name='Samsung A51', buy_price=799, sell_price=909, qty=10,
supplier=User2, sub_category=sub_category2)
```

```
In [36]: User2 = User.objects.get(id=2)

In [37]: User2
Out[37]: <User: Mohammad>

In [38]: sub_category2 = SubCategory.objects.get(id=2)

In [39]: sub_category2
Out[39]: <SubCategory: Sam Phone>

In [40]: product2 = Product.objects.create(name='Samsung A51', buy_price=799, sell_price=909
...: , qty=10, supplier=User2, sub_category=sub_category2)
```

به دنبال محصولاتی هستیم که قیمت آن ها بیشتر از قیمت میانگین کل محصولات باشد.

```
from django.db.models import Avg
```

```
sell_avg = Product.objects.all().aggregate(sell_price_avg=Avg("sell_price"))
```

```
Product.objects.filter(sell_price__gt=sell_avg['sell_price_avg'])
```

```
In [44]: sell_avg = Product.objects.all().aggregate(sell_price_avg=Avg("sell_price"))
```

```
In [45]: Product.objects.filter(sell_price__gt=sell_avg['sell_price_avg'])
```

```
Out[45]: <QuerySet [<Product: Iphone12>, <Product: Samsung edge>]>
```

کاربرانی را پیدا کنید که بیشترین خرید را از ما داشته اند (منظور از خرید بیشترین سبد خرید در وضعیت نهایی یا در صورتی که جدولی به نام **order** دارید بیشترین **order** را داشته اند)

```
from django.db.models import Avg, Count, Max
```

```
count_customer_order = User.objects.annotate(count_order=Count('order'))
```

```
max_order = count_customer_order.aggregate(Max('count_order'))
```

```
count_customer_order.filter(count_order = max_order['count_order__max'])
```

```
In [52]: from django.db.models import Avg, Count, Max
```

```
In [53]: count_customer_order = User.objects.annotate(count_order=Count('order'))
```

```
In [54]: max_order = count_customer_order.aggregate(Max('count_order'))
```

```
In [55]: count_customer_order.filter(count_order = max_order['count_order__max'])
```

```
Out[55]: <QuerySet [<User: Mohammad>]>
```

```
In [56]: max_order
```

```
Out[56]: {'count_order__max': 2}
```

پرفروش ترین کالاهایی که خریداری شده اند را (از زیاد به کم) لیست کنید

```
count_product_sold = Product.objects.annotate(count_item_qty = Sum('orderitem__item_qty'))
```

```
order_count_product_sold = count_product_sold.order_by('-count_item_qty')
```

```
In [65]: count_product_sold = Product.objects.annotate(count_item_qty = Sum('orderitem__item
...: _qty'))

In [66]: count_product_sold
Out[66]: <QuerySet [<Product: Samsung A51>, <Product: Iphone12>, <Product: Samsung A30>, <Pro
duct: Samsung edge>, <Product: Samsung A71s>]>

In [67]: count_product_sold[0].count_item_qty

In [68]: count_product_sold[1].count_item_qty
Out[68]: 4

In [69]: order_count_product_sold = count_product_sold.order_by('-count_item_qty')

In [70]: order_count_product_sold
Out[70]: <QuerySet [<Product: Samsung A30>, <Product: Iphone12>, <Product: Samsung edge>, <Pr
oduct: Samsung A51>, <Product: Samsung A71s>]>

In [71]: order_count_product_sold[0].count_item_qty
Out[71]: 10

In [72]: order_count_product_sold[1].count_item_qty
Out[72]: 4
```

فرض کنید هر کالا دارای یک دسته بندی است میخواهیم پرفروش ترین دسته بندی ها را پیدا کنیم

```
product_sell_in_category = Category.objects.annotate(count_products = Sum('
subcategory__product__orderitem__item_qty')).order_by('-count_products')
```

```
product_sell_in_category[0:3]
```

```
In [193]: product_sell_in_category = Category.objects.annotate(count_products = Sum('
...: subcategory__product__orderitem__item_qty')).order_by('-count_products')

In [194]: product_sell_in_category[0:3]
Out[194]: <QuerySet [<Category: Samsung Co.>, <Category: Apple Co.>, <Category: Nokia
Co.>]>

In [195]: product_sell_in_category[0].count_products
Out[195]: 11

In [196]: product_sell_in_category[1].count_products
Out[196]: 4
```

کاربرانی را پیدا کنید که در روز سوم هفته (مثلا دوشنبه) از ما خرید داشته اند توضیح : مهم این است که در روز دوشنبه از ما خرید داشته ان د مهم نیست کدام دوشنبه.

```
user_which_ordered_mondays = User.objects.filter(order__order_date__week_day=2).distinct()
```

```
user_which_ordered_mondays
```

```
In [205]: user_which_ordered_mondays = User.objects.filter(order__order
...: _date__week_day = 2).distinct()

In [206]: user_which_ordered_mondays
Out[206]: <QuerySet [<User: Mohammad>, <User: lenovo>]>
```

میانگین تعداد تگ هایی که به هر محصول متصل شده اند چقدر است (هر محصول به طور متوسط چند تگ دارد؟)

```
tags_of_product = Product.objects.annotate(count_of_tag=Count('tag'))
```

```
tags_of_product.aggregate(Avg('count_of_tag'))
```

```
In [19]: tags_of_product = Product.objects.annotate(count_of_tag=Count('tag'))

In [20]: tags_of_product.aggregate(Avg('count_of_tag'))
Out[20]: {'count_of_tag__avg': 2.6666666666666665}
```