سوال اول)

منیجر در واقع رابطی است که ارتباط بین عملیات کوئری‌های دیتا بیس و مدل‌های جنگو را برقرار می‌کند. در مدال منیجرهای خواسته شده ساخته شدند.

سوال دوم)

**Multi table inheritance**

برای توضیح این قسمت مثالی که در داکیومنت خود جنگو آمده است گویاست:

from django.db import models

class Place(models.Model):

name = models.CharField(max\_length=50)

address = models.CharField(max\_length=80)

class Restaurant(Place):

serves\_hot\_dogs = models.BooleanField(default=False)

serves\_pizza = models.BooleanField(default=False)

در مثال بالا یک کلاس برای مکان داریم که شامل فیلدهای نام و آدرس است، حال یک کلاس رستوران داریم که از کلاس مکان ارث‌بری می‌کند و علاوه بر آن دو فیلد مربوط به سرو شدن هات داگ و پیتزا که از نوع بولین هستند نیز دارد.

**Proxy Model**

وقتی از مولتی تیبل اینهریتنس استفاده می‌کنیم، برای ذخیره داده‌ها از نوع کلاس فرزند یک جدول جدید ایجاد می‌شود، با استفاده از پروکسی می‌توان در همان جدول قبلی داده‌ را ذخیره کرد و برای داده‌های جدید متودهای اضافی نیز تعریف کرد. باز هم مثال داکیومنت جنگو می‌تواند راهنمای خوبی باشد.

from django.db import models

class Person(models.Model):

first\_name = models.CharField(max\_length=30)

last\_name = models.CharField(max\_length=30)

class MyPerson(Person):

class Meta:

proxy = True

ordering = ["last\_name"]

def do\_something(self):

# ...

pass

در مثال بالا مشاهده می‌شود که یک کلاس به نام شخص داریم که شامل نام و نام خانوادگی است، کلاس شخص من که یک کلاس زیرمجموعه متا دارد که پروکسی در آن ترو قرار داده شده است، این امکان را به وجود می‌آورد که از طریق همان یک جدول به هر دو نوع داده دسترسی داشته باشیم و به علاوه متودهای دیگری نیز به آن اشاره کنیم. کلاس متا یک آیتم اختیاری اردرینگ نیز می‌گیرد که در آن می‌توان مشخص کرد که ترتیب بر اساس کدام فیلد باشد.

سؤال سوم) کوئری اول

from order.models import ListOfCom

from django.db.models import Avg

توضیح: ex\_and\_no منیجر جدید است که به جاری objects تعریف شده است

ListOfCom.ex\_and\_no.filter(price\_\_gt=ListOfCom.ex\_and\_no.all().aggregate(Avg("price"))['price\_\_avg'])

سوال سوم) کوئری دوم

from order.models import Order

from django.db.models import Count

Order.objects.annotate(mc=Count('buyer')).order\_by("-mc")[0].buyer

سوال سوم) کوپری سوم

Order.objects.annotate(mc=Count('comodity')).order\_by("-mc")

سوال سوم) کوپری چهارم

Order.objects.annotate(mc=Count(F("comodity\_\_Category"))).order\_by("-mc")[0].comodity.Category

سوال سوم) کوئری پنجم

Order.objects.filter(order\_date\_\_week\_day=2)

https://docs.djangoproject.com/en/3.2/topics/db/examples/many\_to\_many/

Product.objects.aggregate(Avg=Avg("price\_per\_unit"))

Product.objects.filter(price\_per\_unit\_\_gt=p1["Avg"])

User.objects.annotate(count=Count("salebuy")).order\_by("-count")

Product.objects.annotate(counter = Count("salebuy")).order\_by("-counter")

SubCategory.objects.annotate(count = Count("product\_\_salebuy")).order\_by("-count")

Product.objects.annotate(count = Count("tag"))