

سوال اول)

منیجر در واقع رابطی است که ارتباط بین عملیات کوئری‌های دیتابیس 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 داکيومنت 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 داکيومنت 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.