--------------- model yöneticisi ---------------

Model Yöneticisi (Manager), Django’da veritabanı sorgulama işlemleri gerçekleştirmek için sağlanan arayüzdür. Daha önce her modelin en az bir yöneticisi olduğundan bahsetmiştik. Django varsıyan olarak model yöneticisi olarak objects alan adını kullanır.

**Product.objects**

Objects kullanmak istemiyorsak Model’den değiştirebiliriz.

class Product(models.Model):  
 items = models.Manager()

**product.items.all()**

**product.objects.all()**  # hata ile karşılaşırız.

Modeller için birden fazla yönetici kullanabileceğimiz gibi varsayılan yöneticiyi override da edebiliriz. İlk olarak override edip daha önce Model’de oluşturduğumuz active\_products metodunu model yöneticisine alalım. Active\_productsu @classmethod’a örnek olması açısından model içerisinde yazmıştık normalde model yöneticisinde bulunması daha doğru olur.

class ProductManager(models.Manager):  
 def active\_products(self):  
 return self.filter(active=**1**)

**Products.objects.active\_products()**

Model için birden fazla model yöneticisi de tanımlayabiliriz. Modele iki tane daha yönetici ekleyelim bunlardan bir tanesi ActiveProductManager diğerde PassiveProductManager olsun.

class Product(models.Model):  
 objects = ProductManager()  
 actives = ActiveProductManager()  
 passives = PassiveProductManager()

**Product.objects.all()** # tüm nesneler listelenir

**Product.actives.all()** # aktif nesneler listelenir

**Product.passives.all()** # pasif nesneler listelenir