

به نام خدا

## مدل چیست و چطور به دیتابیس متصل میشود؟

هر مدل معادل یک جدول دیتابیس است و هر شیء از آن معادل سطری از آن جدول خواهد بود. وقتی یک مدل می‌سازیم، Django با استفاده از ORM (Object Relational Mapper) آن را به یک جدول در دیتابیس نگاشت می‌کند.

## نقش migrations در جنگو چیست؟

مهاجرت‌ها (migrations) روشی هستند که برای مدیریت تغییرات در ساختار دیتابیس به صورت نسخه بندی شده. مایگریشن تغییرات مدل را به تغییرات در دیتابیس نگاشت میکند، تضمین میکند که دیتابیس با مدل هماهنگ باشد و تغییرات درتابیس را در تاریخچه ذخیره میکند تا بتوان آن‌ها را اعمال یا حتی به عقب برگرداند.

## چطور میتوانیم یک مدل بسازیم؟

برای ساختن یک مدل کافیت یک کلاس ایجاد کنیم که از `models.Model` ارث‌بری کند و فیلد های مورد نظر را بعنوان ویژگی‌های کلاس تعریف کنیم.

```
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=100)
    published_date = models.DateField()
    price = models.DecimalField(max_digits=6, decimal_places=2)
```

## دستور makemigration چه تفاوتی با دستور migrate دارد؟

دستور makemigration تغییرات مدل را به فایل migration تبدیل میکند و به قولی تغییرات را آماده اعمال در دیتابیس میکند. دستور migrate، فایل های migration را در دیتابیس اجرا میکند.

Django ORM چیست و چه تفاوتی در مقایسه با نوشتن raw SQL ایجاد میکند؟  
 Django ORM ابزاری است که به ما اجازه میدهد به جای نوشتن مستقیم SQL، با استفاده از کلاس ها و اشیاء پایتون با دیتابیس کار کنیم.

♦ تفاوت با Raw SQL		
ویژگی	ORM (Django)	Raw SQL
خوانایی	کد شبیه پایتونه، ساده و قابل فهم	نیاز به دانستن دقیق SQL
وابستگی به دیتابیس	مستقل از دیتابیس (MySQL, PostgreSQL, ... SQLite)	وابسته به نوع دیتابیس و دستورهای خاص اون
امنیت	به طور پیش فرض جلوی SQL Injection رو می گیره	نیاز به احتیاط و استفاده از پارامترهای امن
توسعه سریع	سریع تر در پیاده سازی	نوشتن دستی Query ها وقت گیرتر
انعطاف	برای Query های ساده و متوسط عالیه	برای Query های خیلی پیچیده یا خاص لازم می شه

## تایپ های relationship در Django ؟

سه مدل رابطه در Django models تعریف شده:

**One to One** : مانند زمانی که مشخصات کامل دانشجو در جدول دانشجوها قرار دارد و شماره دانشجو در جدول اعضای کتابخانه.

**One to Many**: مانند زمانی که نام نویسنده به چند رکورد از جدول کتابها نگاشت می شود.

**Many to Many** : مانند زمانی که هر مقاله چندین نویسنده دارد و هر نویسنده چندین مقاله.

در نوع اول، هر رکورد از جدول فقط با یک رکورد در جدولی دیگر می تواند در ارتباط باشد.

علت جداسازی، قرار گیری جداول در سطوح نرمال بالاتر است.

در نوع دوم، هر رکورد از جدول اول می تواند با چند رکورد از جدول دوم در ارتباط باشد اما هر رکورد از

جدول دوم فقط به یک رکورد از جدول اول مرتبط است.

در نوع سوم، هر رکورد از جدول اول می تواند با چند رکورد از جدول دوم در ارتباط باشد و بالعکس.

در این روش، Django به صورت خودکار یک جدول واسطه (junction table) می سازد که نگاشت ID ها را در آن جدول مشخص می کند.

## Django Admin چیست؟

Django Admin یک پنل مدیریتی است آماده است که به صورت خودکار همراه پروژه ایجاد می شود.

برای اینکه مدل ها در پنل ادمین نمایش داده شوند باید آن ها را در فایل admin.py ثبت کنیم.

## نحوه ثبت مدل در پنل ادمین؟

```
from django.contrib import admin
from .models import Book, Author
admin.site.register(Book)
admin.site.register(Author)
```

همچنین می‌توان یک کلاس ModelAdmin تعریف کرد تا ظاهر و رفتار مدل را در پنل ادمین شخصی سازی کنیم:

```
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'author', 'published_date', 'price'),
    search_fields = ('title', 'author'),
    list_filter = ('published_date',)
```

```
admin.site.register(Book, BookAdmin)
```

## Django Form چیست و چه تفاوتی با raw HTML دارد؟

Django form یک کلاس پایتونی است که برای مدیریت فرم ها و ورودی‌های کاربر استفاده می‌شود.

تفاوت با فرم HTML خام (Raw HTML Forms)		
ویژگی	Django Form	Raw HTML Form
تعریف فرم	با کلاس پایتون	با تگ <code>&lt;form&gt;</code> و <code>&lt;input&gt;</code> در HTML
اعتبارسنجی	خودکار، با فیلدها و validators	دستی، معمولاً با جاوااسکریپت یا backend
رندر فرم	می‌توان با <code>{{ form }}</code> یا <code>{{ form.as_p }}</code> مستقیم در template نمایش داد	باید تمام تگ‌ها و <code>name</code> attribute ها را دستی بنویسی
پردازش داده	<code>form.cleaned_data</code> و <code>form.is_valid()</code>	داده‌ها را باید از <code>request.POST</code> یا <code>request.GET</code> پردازش کنی
امنیت	محافظت در برابر CSRF و XSS با یک خط <code>{% csrf_token %}</code>	باید دستی CSRF Token و اعتبارسنجی بنویسی

## مزیت استفاده از Django Form نسبت به raw HTML چیست؟

اعتبارسنجی خودکار:

هر فیلد میتواند نوع داده و محدودیت‌ها را تعریف کند. مثلاً EmailField مطمئن می‌شود داده وارد شده یک ایمیل معتبر باشد.

امنیت:

محافظت در برابر CSRF, XSS, SQL injection .

استفاده مجدد:

فرم‌ها به شکل کلاس تعریف می‌شوند، بنابراین می‌توان همان فرم را در چندین view یا template استفاده کرد.

ادغام با مدل‌ها:

می‌توان فرم‌ها را مستقیماً با مدل‌های Django ادغام کرد.

در این صورت Django به صورت خودکار فیلدها را بر اساس مدل ایجاد می‌کند و validation خودکار اعمال می‌شود.

```
from django.forms import.ModelForm
from .models import Book

class BookForm(ModelForm):
    class Meta:
        model = Book
        fields = ['title', 'author', 'published_date', 'price']
```

تفاوت میان ModelForm و Form در Django ؟

با توجه به جدول زیر، زمانی که نیاز به ساخت فرم سفارشی با پردازش دلخواه داشته باشیم، بهتر است از Form استفاده کنیم.

ویژگی	Form	ModelForm
اتصال به دیتابیس	ندارد	مستقیماً به مدل دیتابیس
تعریف فیلدها	دستی	بر اساس مدل (می‌توان اضافه/حذف کرد)
ذخیره داده‌ها	دستی	با <code>form.save()</code> خودکار
کاربرد	فرم‌های سفارشی، ایمیل، API	فرم‌های CRUD روی مدل‌ها

ارائه دهنده: پویا پورعبداله