

به نام خدا
مهدى فكرى
Task-3

Database:

- What is Relational Database? (for the course we do need to know sql syntax just the whole idea how does relational database work)

دیتابیس رابطه ای دیتا بیسی هست که دیتا هارو به صورت سطر ها و ستون ها ذخیره میکنه و در واقع به صورت جدول ذخیره میکند و ویژگی مهم اینه که بین جدول ها رابطه داریم.

1. What is a model in Django, and how is it connected to the database?

Model ها در جنگو همان نمایش داده ها در قالب پایتون هست و هر مدل در جنگو معادل یک جدول در پایگاه داده است یعنی به طوری که هر کلاس مدل یک جدول در دیتابیس و هر فیلد کلاس یک ستون از جدول و هر اجکت از کلاس مدل هم یک رکورد از جدول رو شام میشه و ارتباط آن با دیتابیس به این طور است که وقتی مدلی رو تعریف میکنیم و migrate میزنیم، جنگو به صورت خودکار اون جدول رو در دیتابیس میسازه و ما میتوانیم با پایتون یه sql به داده ها دسترسی پیدا کنیم.

2. What is the role of migrations in Django?

Migrations در جنگو ابزار هایی هستن که برای اینکه تغییرات مدل ها به شکل خودکار روی خود دیتابیس اعمال شود استفاده میشوند. مثلاً وقتی مدلی رو تغییر میدیم مثلاً یک فیلد جدید اضافه میکنیم با دستور makemigrations اون تغییرات رو ثبت میکنیم و با دستور migrations اون تغییرات رو روی دیتابیس اعمال میکنیم و در واقع migrations پل ارتباطی بین کد پایتون و دیتابیس هستند.

3. How do you create a model class in Django? Provide an example.

برای ساختن model در جنگو کافیه که در فایل model.py یک کلاس بسازیم که از models.model ارث بری میکنه و هر اtribut از کلاس همان یک فیلد از جدول میشه .

From Django.db import models

Class student(models.Model):

Name = models.CharField(max_length = 100)

StudentNumber = models.IntegerField()

4. What is the difference between makemigrations and migrate?

Makemigrations میاد تغییرات رو بررسی میکنه و یک فایل migration میسازه که میگه باید چه تغییراتی روی دیتابیس اعمال بشه در حالی که migrate میاد اون فایل migration رو اجرا میکنه و واقعا ساختار دیتابیس رو تغییر میده.

5. What is the Django ORM, and how does it differ from writing raw SQL queries?

ORM یا همام object relational mapper به ابزاری هست که به ما اجازه میده به جاس نوشتمن مستقیم sql با کد پایتون با دیتابیس تعامل کنیم. و از جمله تفاوت های آن به sql میتوان به اینکه در ORM همه چیز رو با کلاس و آجکت های پایتون مدیریت میکنیم و برای یک برنامه نویس پایتون خواناتر و ساده تر است و از نوع دیتابیس مستقل است در حالی که sql باید مسلط به کوئری نوشتمن باشیم و قدرت و انعطاف بیشتری برای کوئری های پیچیده دارد.

6. How do you perform basic queries with the ORM? For example:

- o Create a new object and save it.

```
Object = Classname(fields)
```

```
Object.save()
```

Or

```
Object = Classname.object.create(fields)
```

- o Retrieve all objects.

```
Objects = Classname.objects.all()
```

- o Filter objects by a condition.

```
Filtered_object = Classname.objects.filter(condition)
```

- o Update an object.

```
Object = Classname.objects.get(condition)
```

```
Object.attr = new value
```

```
Object.save()
```

- o Delete an object.

```
Object = Classname.objects.get(condition)
```

```
Object.delete()
```

7. What are the three types of relationships in Django models?

- o One-to-One (why not just use one model?)

- o One-to-Many (ForeignKey)
- o Many-to-Many (what happened in behind)

خب تو این سوال درباره رابطه ها در مدل صحبت میکنیم

رابطه یک به یک به این صورت است که هر رکورد در یک جدول دقیقاً با یک رکورد در جدول دیگر مرتبط است مثلاً هر شخصی یک حساب کاربری دارد و هر حساب کاربری مرتبط به یک کاربر است یا مثلاً رابطه کشور با پایتخت هم که یک به یک است و دلیل اینکه از یک جدول استفاده نمیکنیم و از دو تا مدل استفاده نمیکنیم به این دلیل است که جدول شلوغ تر نشود و ممکنه برای یک مقدار، مقداری اختیاری داشته باشیم.

رابطه بعدی رابطه یک به چند است که رابطه کشور با شهر و مثلاً استاد با دانشکده میشود که در این صورت از کلید خارجی استفاده میکنیم و کلید سمت یک رو به عنوان فیلد به فیلد های سمت چند اضافه میکنیم.

رابطه بعدی رابطه چند به چند است که مثلاً درس با دانشجوکه هر دانشجو میتوانه چند درس و هر درس میتوانه چند دانشجو متعلق باشد که این رابطه تنها رابطه ای است که یک جدول دیگه ایجاد میکند. مثلاً همان بخواهیم رابطه دانشجو با درس رو مثال بزنم یک جدول دیگه هم به اسم takes ایجاد میشود که کلید اصلی هر دو طرف اضافه میشود.

8. Provide an example of when you would use each relationship.

جواب این سوال یا همان مثال برای هر کدام از روابط بالا را در همان سوال قبلی توضیح دادم.

9. How do you define relationships in Django models? Show code snippets.

```
:class User(models.Model)
```

```
username = models.CharField(max_length=100)
```

```
:class Profile(models.Model)
```

```
user = models.OneToOneField(User, on_delete=models.CASCADE)
```

```
()bio = models.TextField
```

```
:class Teacher(models.Model)
```

```
name = models.CharField(max_length=100)
```

```
:class Student(models.Model)
```

```
name = models.CharField(max_length=100)
```

```
teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
```

```
:class Course(models.Model)  
title = models.CharField(max_length=100)  
  
:class Student(models.Model)  
name = models.CharField(max_length=100)  
courses = models.ManyToManyField(Course)
```

Admin and Data Management

10. What is Django Admin?

جنگو ادمین یک بخش آماده هستش که بعد از ساخت پروژه به صورت خودکار فعال میشه و یک پنل مدیریتی تحت وب برای مدیریت داده ها رو بر عهده میگیره و اجازه عمل های CRUD رو به ما میده.

11. How do you register models in the Django admin site?

برای اینکه یک مدل در جنگو ادمین نمایش داده بشه کافیه اونو تو فایل admin.py اضافه کنیم

```
admin.site.register(Student)
```

12. How do you create a superuser and log in to the Django admin?

```
python manage.py createsuperuser  
python manage.py runserver  
http://127.0.0.1:8000/admin/
```

Forms

9. What is a Django Form, and how does it differ from raw HTML forms?

جنگو فرم یک کلاس در جنگو هستش که برای ساخت و اعتبار سنجی داده ها استفاده میشود و به جای اینکه همه چیز رو با HTML بنویسیم ، جنگو بخش بزرگی از کار رو خودکار انجام میده و میتوان از تفاوت های آن با HTML خام به این اشاره کرد که اولاً امنیت بیشتری داره و اینکه میتوانه به صورت خودکار کد HTML رو بسازه و فیلد هارو هم به صورت خودکار میسازه و همچنین با modelform میتوان با دیتابیس وصل بشیم ولی در HTML خام نمیتوانیم این کار رو انجام بدیم.

10. How do you create a forms.py file and connect it to a view?

خب اول یک فایل میسازیم در داخل آن یک کلاس فرم با فیلد های مورد نظر درست میکنیم و سپس در view از آن استفاده میکنیم. یعنی در forms.py فرم رو تعریف میکنیم و در views.py پردازش میکنیم.

11. What are the advantages of Django forms (e.g., validation, security, reusability)?

این رو در سوال 9 توضیح دادم.

12. What's the difference between ModelForm and Form in Django? When would you use

each?

در فرم باید دستی همه فیلد هارو تعریف کنیم و ارتباط مستقیم با دیتابیس نداریم و مناسب وقته است که میخواهیم داده هارو وقت پردازش کنیم وای مدل فرم مستقیم به مدل وصل میشه و فیلد های خودش رو از روس مدل درست میکنه .