

مستندات مدل ها و دیتابیس پروژه مدیریت پروژه های دانشجویی

مریم محمدآبادی پور

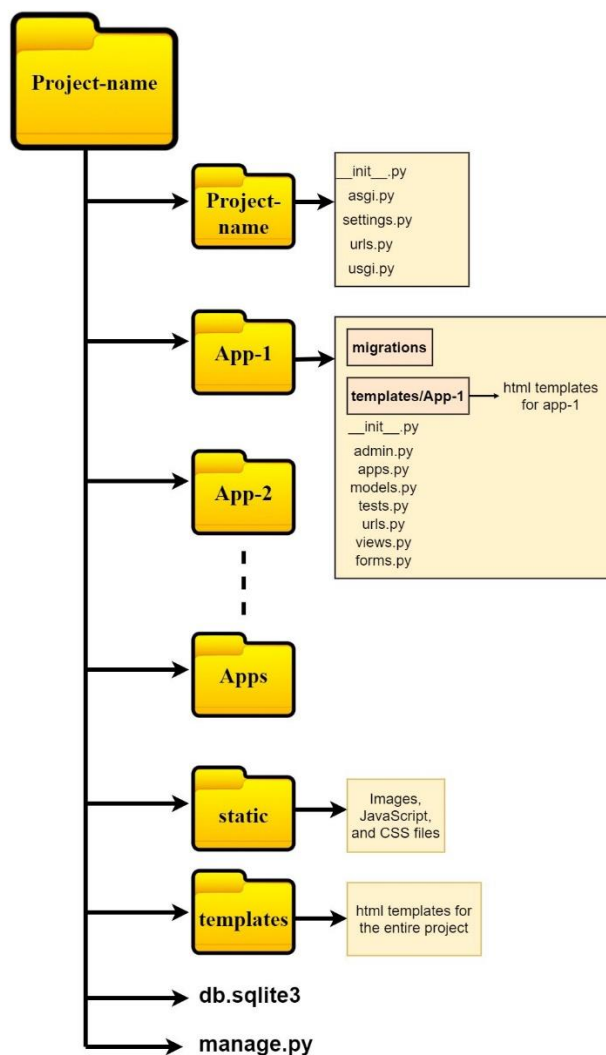
در تصویر روبرو مدل کلی فایل های جنگو نمایش داده شده است:

- منظور از App <برنامه های کاربردی> (Applications) است.

- فایل **__init__.py** این فایل توسط کتابخانه جنگو تولید می شود تا اطمینان حاصل شود که مفسر پایتون، این پوشه را به عنوان یک بسته برنامه نویسی (Package) معتبر در نظر و برای مقاصد بسته سازی برنامه کاربردی مورد استفاده قرار می گیرد.

- فایل **admin.py** این فایل به برنامه نویس و توسعه دهنده وب اجازه می دهد تا بتواند برنامه کاربردی ایجاد شده را در «واسط مدیریتی» (Administrator Interface) تغییر دهد.

- فایل **models.py** مکانی است که در آن تمام مدل های برنامه کاربردی ذخیره خواهند شد.



- فایل **tests.py** مکانی است که تنظیمات مرتبط با ارزیابی و تست واحدها یا مؤلفه‌های برنامه کاربردی در آن قرار می‌گیرد.

- فایل **views.py** مکانی است که تنظیمات و کدهای مرتبط با View برنامه کاربردی در آن قرار می‌گیرد.

- فایل **manage.py** این فایل به نحوی نقطه ایجاد تعامل با پروژه تولید شده در واسط خط دستور است) راه‌اندازی «سرور توسعه (Development Server)»، همگام‌سازی پایگاه داده و سایر موارد. (برای نمایش لیست کامل دستورات قابل دسترس از طریق فایل **manage.py**، کافی است دستور زیر را اجرا کنید:

```
python manage.py help
```

App های پروژه :

۱. Accounts

۲. Administration

۳. Students

۴. Teachers

در ادامه مدل ها براساس ترتیب ایجاد شدن شرح داده خواهد شد.

Administration\models

administration > models.py > Project > __str__

```
1 from django.db import models
2 from django.contrib.auth.models import User
3
4 # Create your models here.
5
6 PROJECT_TYPE_CHOICES = (
7     ("web", "تحت وب"),
8     ("desktop", "دسکتاپ"),
9     ("mobile", "موبایل"),
10 )
11
12 PROJECT_MEMBER_CHOICES = (
13     (1, "یک"),
14     (2, "دو"),
15     (3, "سه"),
16 )
17
18
19 class Project(models.Model):
20     title = models.CharField(max_length=255)
21     creator = models.ForeignKey(User, on_delete=models.CASCADE)
22     type = models.CharField(max_length=150, choices=PROJECT_TYPE_CHOICES)
23     number_of_students = models.IntegerField(choices=PROJECT_MEMBER_CHOICES)
24     description = models.TextField()
25     is_suggested = models.BooleanField(default=False)
26
27     def __str__(self):
28         return self.title
29
```

این مدل برای ساخت جدول پروژه‌ها
ایجاد شده است.

تمام پروژه‌های ایجاد شده توسط استاد یا پروژه
های پیشنهاد شده توسط دانشجویان که توسط
استاد قبول شده اند در این جدول قرار میگیرند.

توضیح کد:

ابتدا دو تاپل تعریف شده که تاپل اول انتخاب‌های
موجود برای تایپ پروژه را در نظر گرفته و تاپل دوم
تعداد ظرفیت هر پروژه است.

* اتریبیوت‌ها در جدول صفحه بعد توضیح داده
شده است.

و در آخر متد `str` باعث میشود وقتی یک شی از پروجکت داشته باشیم عنوان (`title`) آن را نمایش دهد.

کلید ها عبارتند از:

Project

Key name	title	creator	type	number_of_students	description	is_suggested
Data Type	CharField	کلید خارجی user به	CharField	IntegerField	TextField	BooleanField
توضیحات	عنوان پروژه		و انتخاب هایی که بهش دادیم تاپل CHOISES_TYPE_PROJECT (وب،دستکپتاب و موبایل)را داشتیم	ظرفیت پروژه از یک تاپل استفاده شده است و محدوده انتخاب ظرفیت از ۱ تا ۳ است.	توضیحات	ایا انتخاب شده است توسط دانشجویان

Accounts\models

accounts > models.py > ...

```
1 from django.db import models
2 from django.contrib.auth.models import User
3 from administration.models import Project
4
5 # Create your models here.
6
7
8 class CollegeUsers(models.Model):
9     user = models.OneToOneField(User, on_delete=models.CASCADE)
10     firstname = models.CharField(max_length=150)
11     lastname = models.CharField(max_length=150)
12     project = models.ForeignKey(Project, models.SET_NULL, null=True, blank=True)
13     is_teacher = models.BooleanField(default=False)
14     is_student = models.BooleanField(default=True)
15
16     def __str__(self):
17         return f"{self.firstname} {self.lastname}"
18
```

این مدل برای ساخت جدول collegeUsers نوشته شده است.

برای تمامی یوزر هایی که داخل سایت وجود دارند.

ادمین .اساتید و دانشجویان

از مدل اپ ادمین استفاده شده است پس لازم است در ابتدا ایمپورت شود.

این کلاس برای این است که یوزر جنگو را گسترش بدهد و فیلدهای مورد نظر را اضافه کند باید یک وان تو وا فیل (OneToOneField) ایجاد کند به یوزر جنگو

بنابراین یک فیلد به اسم user را ایجاد کردیم و یک OneToOneField که به یوزر وصل میشود و CASCADE.models=delete_on

یعنی اگر یوزر جنگو حذف بشود ،ان یوزی که ایجاد شده که به CollegeUsers وصله حذف نمیشود و باقی میماند.
*فیلد ها در جدول صفحه بعد توضیح داده شده است.

متد str نام (firstname) و نام خانوادگی (lastname) یوزر را برمیگرداند.

CollegeUsers

Key name	user	firstname	lastname	project	is_teacher	is_student
Data Type	OneToOneField	CharField	CharField	کلید خارجی project به FOREIGN KEY	BooleanField	BooleanField
توضیحات	یک OneToOneField ایجاد کند به یوزر جنگو	نام یوزر	نام خانوادگی		ایا استاد است؟	ایا دانشجو است؟

Students\models

students >  models.py

```
1  from django.db import models
2
3  # Create your models here.
4  
```

...

Teachers\models

teachers > models.py > ...

```
1 from django.db import models
2 from accounts.models import CollegeUsers
3 from administration.models import Project
4
5 # Create your models here.
6
7
8 class ProjectSetting(models.Model):
9     can_get = models.BooleanField()
10    suggest = models.BooleanField()
11    reports = models.BooleanField()
12    final_report = models.BooleanField()
13
14    def __str__(self):
15        | return "برای تغییر ضربه بزنید"
16
17
18 class ReportParts(models.Model):
19     number = models.IntegerField()
20     is_active = models.BooleanField(default=False)
21
22     def __str__(self):
23         | return f"Part {self.number}"
24
25
26 class ProjectReport(models.Model):
27     student = models.ForeignKey(CollegeUsers, on_delete=models.CASCADE)
28     number = models.IntegerField()
29     file = models.FileField(null=True, blank=True)
30     description = models.TextField(null=True, blank=True)
31     mark = models.IntegerField(null=True, blank=True)
32     date = models.DateField(auto_now=True)
33     is_submit = models.BooleanField(default=True)
34
35     def __str__(self):
36         | return f"{self.student} - {self.number}"
```

ابتدا models از db.Django و

مدل کلاس collegeusers در اپ accounts و

مدل کلاس project را اپ administration ایمپورت شده است.

سپس در این فایل ۴ تا مدل تعریف کرده ایم:

- ProjectSetting
- ReportParts
- ProjectReport
- FinalReport

ProjectSetting

teachers > models.py > ...

```
1 from django.db import models
2 from accounts.models import CollegeUsers
3 from administration.models import Project
4
5 # Create your models here.
6
7
8 class ProjectSetting(models.Model):
9     can_get = models.BooleanField()
10    suggest = models.BooleanField()
11    reports = models.BooleanField()
12    final_report = models.BooleanField()
13
14    def __str__(self):
15        return "برای تغییر ضربه بزنید"
16
17
```

تنظیمات پروژه را در قالب یک کلاس در نظر گرفتیم .
چهارتا فیلد بولین داریم که در جدول صفحه بعد شرح
داده شده اند.

متد **str** هم ریترن میکند یک رشته با این عنوان " برای تغییر ضربه بزنید "
متد **str** در ادمین سایت جنگو استفاده میشود.

ProjectSetting

Key name	can_get	suggest	reports	final_report
Data Type	BooleanField	BooleanField	BooleanField	BooleanField
توضیحات	ایا میتوانیم ان را اخذ کرد	ایا پیشنهاد شده است	گزارشات وجود دارد یا ن	گزارش نهایی وجود دارد یا ن

ReportParts

این کلاس برای طراحی جدول مربوط به تعداد گزارش ها تعیین شده است.

به این صورت که **number** شماره گزارش مورد نظر است (از ۱ تا ۸) و **is_active** برای فعال بودن یا نبودن گزارش است. که استاد میتواند هر گزارشی که خواست را فعال یا غیرفعال کند.

ReportParts

Key name	number	is_active
Data Type	IntegerField	BooleanField
توضیحات	شماره گزارش	ایا فعال است یا خیر

ProjectReport

```
26 class ProjectReport(models.Model):
27     student = models.ForeignKey(CollegeUsers, on_delete=models.CASCADE)
28     number = models.IntegerField()
29     file = models.FileField(null=True, blank=True)
30     description = models.TextField(null=True, blank=True)
31     mark = models.IntegerField(null=True, blank=True)
32     date = models.DateField(auto_now=True)
33     is_submit = models.BooleanField(default=True)
34
35     def __str__(self):
36         return f"{self.student} - {self.number}"
37
```

این مدل برای جزییات گزارش های ارسالی توسط دانشجویان طراحی شده است.

• یک فیلد به اسم student ایجاد شده که فارن کی است به CollegeUsers به این صورت است که هر پروجکت ریپورتی یک student دارد و هر دانشجو میتواند چندتا گزارش یا ProjectReport داشته باشد .

آن_دلایت هم cascade.model است.یعنی اگر این کالج یوزر حذف شد پروجکت ریپورت باقی بماند.

متد str هم student را برمیگرداند

*student یک شی از CollegeUsers است که فرست نیم و لست نیم ان برگردونده میشود از str و اینجا نامبر هم قرار دادیم و برگردانده میشود.

ProjectReport

Key name	student	number	file	description	mark	date	is_submit
Data Type	کلید خارجی collegeusers به	Integer Field	File Field	Text Field	Integer Field	Date Field	Boolean Field
توضیحات	به این صورت است که هر پروجکت ریپورتی یک student دارد و هر دانشجو میتواند چندتا گزارش یا ProjectReport داشته باشد. آن_دلیت هم cascade.model است. یعنی اگر این کالج یوزر حذف شد پروجکت ریپورت باقی بماند		برای فایل اپلودی توسط دانشجو	توضیحاتی که دانشجو برای گزارش مینویسد	نمره	تاریخ	یک بولین است که دیفالت True و میگه که این گزارش تایید شده است یا خیر

FinalReport

این مدل برای گزارش نهایی هر پروژه ایجاد شده است.

```
38 |
39 class FinalReport(models.Model):
40     student = models.ForeignKey(CollegeUsers, on_delete=models.CASCADE)
41     project = models.ForeignKey(Project, on_delete=models.SET_NULL, null=True)
42     file = models.FileField(null=True, blank=True)
43     description = models.TextField(null=True, blank=True)
44     mark = models.IntegerField(null=True, blank=True)
45     is_archive = models.BooleanField(default=False)
46
47     def __str__(self):
48         return f"{self.student}"
49
```

متد str برمیگرداند student.self که خودش یک متد str دارد که لست نیم و فرست نیم برمیگرداند

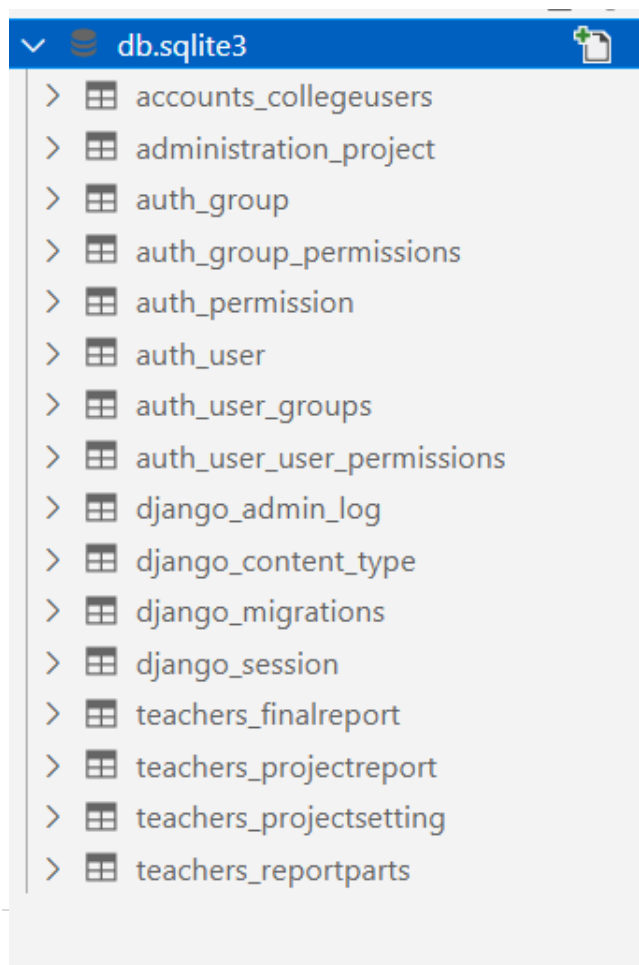
FinalReport

Key name	student	project	file	description	mark	is_archive
Data Type	کلید خارجی به CollegeUsers	کلید خارجی بهproject	File Field	TextField	Integer Field	Boolean Field
توضیحات	on_delete: اگ استیودنت حذف شد فاینال ریپورت را حذف کن	on_delete : اگ یک پروژه ای حذف شد فاینال پروژه را نال کن. null=true	اپشنال است چون نال و بلنک true است	اپشنال است (اختیاری)	اپشنال	دیفالتش False است. یعنی در ابتدا غیرفعال است.

DataBase

در این پروژه من از دیتابیس پیشفرض جنگو (sqlite) استفاده کرده ام.
این دیتابیس برای توسعه خوب است.

به دلیل اینکه پروژه من آنلاین نیست به نظرم این دیتابیس مناسب بود.
در ادامه توضیحاتی درمورد این دیتابیس شرح داده ام.



accounts_collegeusers جدول

برای تمامی یوزرها

SQLITE EXPLORER

db.sqlite3

accounts_collegeusers

- id : integer
- firstname : varchar(150)
- lastname : varchar(150)
- is_teacher : bool
- is_student : bool
- user_id : integer
- project_id : bigint

administration_project

auth_group

auth_group_permissions

auth_permission

SQL ▼

< 1 / 1 > 1 - 7 of 7

id	firstname	lastname	is_teacher	is_student	user_id	project_id
1	مریم	محمد ابادی پور	0	0	1	NULL
2	زهرا	محمدی	1	0	2	NULL
3	سجاد	برهانی	1	0	3	NULL
4	فاطمه	رضایی	0	1	4	1
5	علی	زاهدی	0	1	5	2
6	مریم	برهانی	0	1	6	4
8	فاطمه	عطایی	1	0	8	NULL

administration_project جدول

برای همه ی پروژه ها

SQLITE EXPLORER

db.sqlite3

accounts_collegeusers

administration_project

- id : integer
- title : varchar(255)
- type : varchar(150)
- number_of_students : integer
- description : text
- is_suggested : bool
- creator_id : integer

auth_group

auth_group_permissions

auth_permission

auth_user

auth_user_groups

auth_user_user_permissions

django_admin_log

django_content_type

django_migrations

django_session

teachers_finalreport

teachers_projectreport

teachers_projectsetting

teachers_reportparts

- id : integer
- is_active : bool
- number : integer

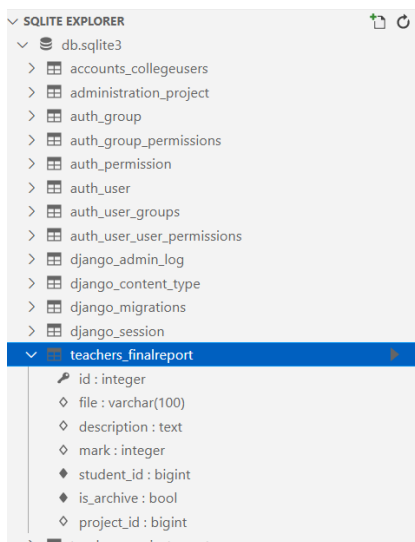
SQL ▼

< 1 / 1 > 1 - 10 of 10

id	title	type	number_of_students	description	is_suggested	creator_id
1	اپلیکیشن مترجم	mobile	2	توضیحات اپلیکیشن مترجم	0	2
2	وبسایت رستوران	web	1		0	2
3	بازی شطرنج	mobile	3		0	2
4	ماشین حساب	mobile	1		0	2
5	وب سایت مشاوره	web	1		1	4
6	وب سایت موسسه آموزشی	web	1		1	4
7	طراحی وب سایت خبری	web	3		0	8
8	پروژه وب پخش آنلاین ویدیو و موزیک	web	2		0	8
9	طراحی وبلاگ	web	1		1	6
10	طراحی سایت فروشگاه (پیشنهاد دانشجو)	web	1		1	6

جدول teachers_finalreport

برای گزارش های نهایی



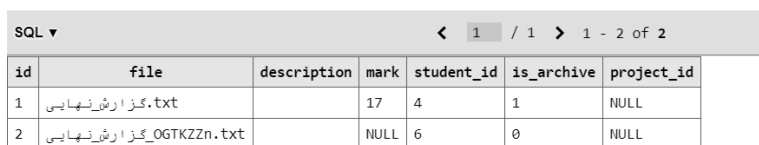
SQLite Explorer

db.sqlite3

- accounts_collegeusers
- administration_project
- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- teachers_finalreport**
- teachers_projectreport

teachers_finalreport fields:

- id : integer
- file : varchar(100)
- description : text
- mark : integer
- student_id : bigint
- is_archive : bool
- project_id : bigint



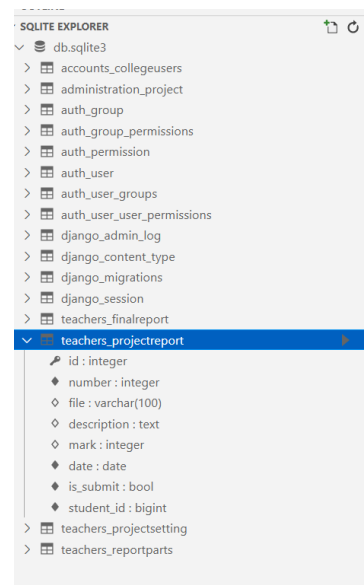
SQL

1 / 1 1 - 2 of 2

id	file	description	mark	student_id	is_archive	project_id
1	گزارش نهایی.txt		17	4	1	NULL
2	گزارش نهایی_OGTKZZn.txt		NULL	6	0	NULL

جدول teachers_projectreport

برای گزارش های ارسال شده در طول ترم



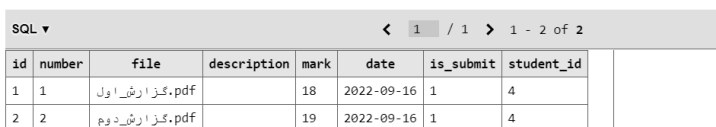
SQLite Explorer

db.sqlite3

- accounts_collegeusers
- administration_project
- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- teachers_finalreport
- teachers_projectreport**
- teachers_projectsetting
- teachers_reportparts

teachers_projectreport fields:

- id : integer
- number : integer
- file : varchar(100)
- description : text
- mark : integer
- date : date
- is_submit : bool
- student_id : bigint



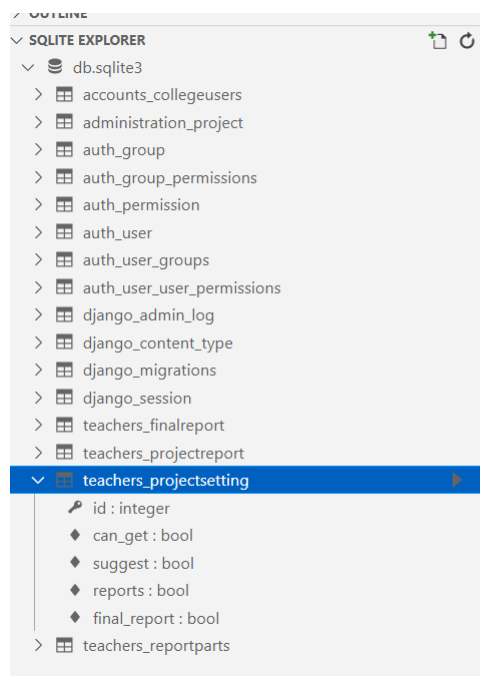
SQL

1 / 1 1 - 2 of 2

id	number	file	description	mark	date	is_submit	student_id
1	1	گزارش اول.pdf		18	2022-09-16	1	4
2	2	گزارش دوم.pdf		19	2022-09-16	1	4

جدول teachers_projectsetting

برای تنظیمات هر پروژه



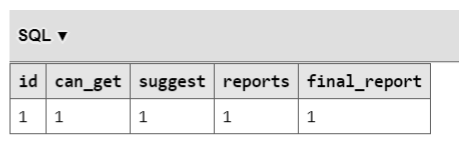
SQLite Explorer

db.sqlite3

- accounts_collegeusers
- administration_project
- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- teachers_finalreport
- teachers_projectreport
- teachers_projectsetting**
- teachers_reportparts

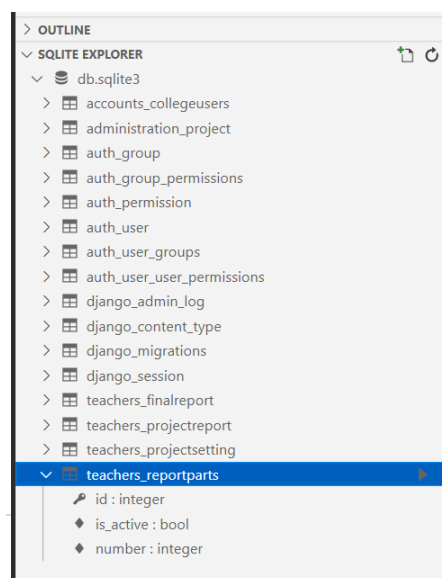
teachers_projectsetting

- id : integer
- can_get : bool
- suggest : bool
- reports : bool
- final_report : bool



SQL

id	can_get	suggest	reports	final_report
1	1	1	1	1



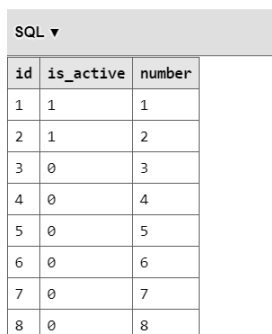
SQLite Explorer

db.sqlite3

- accounts_collegeusers
- administration_project
- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- teachers_finalreport
- teachers_projectreport
- teachers_projectsetting
- teachers_reportparts**

teachers_reportparts

- id : integer
- is_active : bool
- number : integer



SQL

id	is_active	number
1	1	1
2	1	2
3	0	3
4	0	4
5	0	5
6	0	6
7	0	7
8	0	8

جدول teachers_reportparts

اطلاعات ۸ تا گزارش تعریف شده که استاد میتواند انها را فعال یا غیرفعال کند.

DataBase

تمام اطلاعات سایت از قبیل متن، تصاویر، تغییرات انجام گرفته بر روی تنظیمات اصلی سایت، اطلاعات مربوط به افزونه ها و... بر روی دیتابیس سایت ذخیره می شوند.

دیتابیس ها از ساختار SQL یا Structured Query Language بهره مند هستند که به ما این امکان را می دهد تا اطلاعات گسترده و انبوهی را به صورت طبقه بندی شده در آن ها ذخیره کنیم و به راحتی آن ها را بیابیم. امروزه تقریبا تمامی وب سایت های داینامیک که در داخل کشور طراحی و توسعه می یابند، از یک دیتابیس بهره مند هستند. در وب سایت های مدرن امروزی، فایل ها صرفا شامل کدهای PHP, ASP.Net, Java و .. هستند که در طول مدت زمان تغییر چندان هم نمی کنند و هربار اطلاعاتی را از دیتابیس خوانده و به کاربر نمایش می دهد. لذا اطلاعات موجود در دیتابیس با توجه به اینکه دائما در حال تغییر هستند، بسیار حساس شده و بک آپ گیری و مدیریت مناسب آن ها در فضای هاست و سرور از اهمیت بالایی برخوردار است.

فریمورک جنگو از ORM یا همون Object Relational Model استفاده میکنه .

برای اینکه بخوایم شمای دیتابیس مون رو برای استفاده طراحی کنیم، باید بیایم و data model بسازیم.

Model درحقیقت یه کلاس پایتون هستش که از زیر مجموعه های `django.db.models.Model` هست که توی اون هر اتریبیوت، نمایشگر یه فیلد دیتابیس.

جنگو برای هر مدلی که توی فایل `models.py` تعریف شده باشه بصورت خودکار به جدول توی دیتابیس میسازه.

وقتی هم که مدل مورد نظر را ساختیم، دیگه به راحتی میشه توسط سیستم ORM جنگو، اقدامات مورد نیاز را روی دیتابیس انجام بدیم.

بعد از تعریف مدل ها لازم است که **data model** ها رو به جداول دیتابیس تبدیل کنیم. جنگو برای این کار، یه سیستم **migration** داره که همه تغییراتی که روی مدل ها اعمال کردیم رو دنبال میکنه و آماده شون میکنه تا بتونیم ببریمشون توی دیتابیس.

جنگو در دایرکتوری **app name/migrations** یه فایل با اسم **۰۰۰۱_initial.py** میسازد که عملیات سینک کردن دیتابیس با مدل های ما توسط کدهای موجود در این فایل انجام میشود.

SQLite چیست؟

SQLite یک سیستم مدیریت پایگاه داده رابطه‌ای، Embedded و متن-باز است که در حدود سال ۲۰۰۰ طراحی شده است. SQLite یک دیتابیس سبک و بدون نیاز پیکربندی است که برای اجرا شدن به سرور و یا نصب نیاز ندارد. دیتابیس SQLite علی‌رغم سادگی خود، مجهز به قابلیت‌های رایج بسیاری از سیستم‌های مدیریت دیتابیس دیگر است.

SQLite پرکاربردترین دیتابیس در جهان است، از آن به عنوان یک کتابخانه در میان دیگر فرایندهای برنامه‌نویسی می‌توان استفاده کرد و کدهای آن در دسترس عموم قرار دارد. برای استفاده به هر منظور، تجاری یا خصوصی رایگان است. در واقع آن را می‌توان یک دیتابیس embedded از نوع SQL دانست.

خصوصیات کلیدی SQLite

SQLite نسبت به سایر سیستم‌های مدیریت پایگاه داده مانند SQL Server یا Oracle بسیار حجم کمتری دارد (حجم فایل آن از ۵۰۰ کیلوبایت کمتر است).

SQLite یک سیستم مدیریت پایگاه داده بر اساس رابطه کلاینت-سرور نیست، بلکه کتابخانه درون حافظه‌ای است که می‌توانید بدون نصب و پیکربندی مستقیماً آن را فراخوانی کرده و مورد استفاده قرار دهید

یک دیتابیس SQLite به طور معمول شامل یک فایل منفرد است که به همراه همه اجزای دیتابیس مانند جداول، نماها، تریگرها و غیره روی فضای دیسک رایانه شما ذخیره می‌شود. بدین ترتیب هیچ نیازی به وجود یک سرور اختصاصی وجود ندارد.

چه زمانی باید از SQLite استفاده کنیم؟

SQLite می‌تواند حجم درخواست‌های پایین تا متوسط HTTP را اداره کرده و اطلاعات پیچیده نشست‌ها را برای یک وب‌سایت مدیریت کند.

زمانی که نیاز باشد یک آرشیو از فایل‌ها داشته باشیم، SQLite می‌تواند آرشیوها با اندازه کوچک‌تر و با متادیتای کمتر نسبت به آرشیوهای معمول ZIP تولید کند.

SQLite یک روش آسان و کارآمد برای پردازش داده‌ها با استفاده از متغیرهای درون حافظه‌ای در اختیار شما قرار می‌دهد.

زمانی که به یک سیستم دیتابیس برای یادگیری و آموزش نیاز دارید، SQLite گزینه مناسبی محسوب می‌شود. چنان که پیش‌تر اشاره کردیم، این دیتابیس هیچ نیازی به نصب و پیکربندی ندارد. کافی است کتابخانه SQLite را روی رایانه خود کپی کنید تا شروع به یادگیری آن نمایید.

ضرورت آموزش دیتابیس SQLite چیست؟

SQLite رایگان است. این دیتابیس به صورت متن-باز ارائه شده است و هیچ لایسنس تجاری برای کار با آن مورد نیاز نیست.

SQLite یک سیستم مدیریت دیتابیس چند پلتفرمی است. می توان از این سیستم روی طیف وسیعی از پلتفرم ها مانند ویندوز، مک، لینوکس و یونیکس استفاده کرد. همچنین می توان از SQLite روی انواع مختلفی از سیستم های عامل embed-شده مانند Symbian و Windows CE استفاده کرد.

SQLite یک روش مؤثر برای ذخیره سازی داده ها ارائه می کند، چون در آن طول ستون ها متغیر بوده و ثابت نیست. از این رو SQLite تنها فضایی که نیاز دارد را تخصیص می دهد. برای نمونه اگر یک ستون varchar (۲۰۰) داشته باشید، و یک مقدار با طول ۱۰ کاراکتر در آن قرار دهید، varchar (۲۰۰) تنها فضای ۲۰ کاراکتر را برای آن فیلد اختصاص می دهد و دیگر ۲۰۰ کاراکتر اشغال نمی کند.

طیف وسیعی از API-های SQLite وجود دارند. SQLite برخی API-ها برای انواع مختلفی از زبان‌های برنامه‌نویسی ارائه می‌کند. از جمله این زبان‌ها شامل زبان‌های NET مانند ویژوال بیسیک، سی شارپ، PHP، جاوا، آجکتیو C، پایتون و بسیاری از زبان‌های دیگر برنامه‌نویسی است. SQLite بسیار انعطاف‌پذیر است.

متغیرهای SQLite دارای نوع‌بندی دینامیک هستند، یعنی نوع متغیر تا زمانی که مقداری به آن انتساب نیافته است، مشخص نمی‌شود و در زمان اعلان تعریف نشده است.

با استفاده از گزاره `INSERT ON CONFLICT REPLACE` می‌توان به SQLite اعلام کرد که باید یک درج روی جدول انجام دهد و اگر ردیف‌هایی با کلیدهای اصلی یکسان بیابد، در این صورت باید آن مقادیر را با مقادیر درج شده به‌روزرسانی کند.

با استفاده از SQLite می‌توانید هم‌زمان و در یک نشست روی چندین دیتابیس کار کنید. کافی است این دیتابیس‌ها را الحاق کنید تا به همه اجزای این دیتابیس‌ها شامل جداول، نماها و غیره دسترسی داشته باشید.

SQLite

یک دیتابیس با SQL engine است که با ویژگی‌های جامع بودن، قابل اطمینان بودن، embedded یا جاسازی شده، دارای قابلیت استفاده در حوزه‌های عمومی و ... ارائه شده است.

خواندن و نوشتن در فایل‌های دیسک را به راحتی انجام می‌دهد زیرا هیچگونه سرور جداگانه‌ای مانند SQL ندارد. فرمت فایل‌های SQLite به صورت cross-platform یا چند سکویی است و داده‌های آن را به راحتی می‌توان بین سیستم‌های ۳۲ و ۶۴ بیتی کپی کرد. با توجه به این ویژگی‌ها، این دیتابیس یک انتخاب محبوب بین برنامه‌هایی است که بر اساس فرمت فایل کار می‌کنند.