

Django

models



План

- что есть модель и ее поля и связь с БД
- еще раз про миграции
- типы полей, атрибуты полей
- связи между таблицами
- кверисеты, фильтрация и типы фильтров
- создание-редактирование-удаление
- отображение виджетов в админке



models

- INSTALLED_APPS
- python manage.py makemigrations
- python manage.py migrate
- python manage.py dbshell

Каждая модель в приложениях является описанием и способом связаться с соответствующей таблицей в БД.

Каждое поле в модели - это описание столбца в таблице



models - fields

В БД

- название поля - название столбца в БД
- тип поля - тип столбца в БД
- атрибуты поля - дополнительные параметры столбца

В django

- название поля - обращаемся по имени в коде
- тип поля - определяет виджеты в админке и формах
- атрибуты поля - дополнительные параметры виджетов и валидации



models - fields

Числа - INTEGER

AutoField

IntegerField

Строки - VARCHAR

CharField

EmailField

SlugField

FileField

URLField

Текстовые поля - TEXT

TextField

Булевое - BOOL

BooleanField

Дата-время - DATE

DateField

DateTimeField

TimeField



models - fields - options

null (True/False)

Допустимы ли пустые значения (NULL) в БД

blank (True/False)

Можно ли сохранять пустую строку (на уровне валидации)

choices

Ограничивает набор хранимых в поле значений

default

Значение по умолчанию



models - fields - options

help_text

Справочная строка

unique (True/False)

Признак уникальности значения в поле

verbose_name

Можно задать название полей которое будет выводиться в админе и формах



models - relationships

M-to-1

Многие к одному или внешний ключ
Поле **ForeignKey**

Суть в том что одна запись одной таблицы относится к конкретной записи из другой таблицы: студент и группа, вопрос и ответы, пост и комментарии...

В группе может быть много студентов, но студент может быть только в одной группе.

На уровне БД просто создается поле в котором хранится id связанного объекта



models - relationships

1-to-1

Один к одному

Поле `OneToOneField`

Один объект просто дополняет другой своими свойствами: студент и его парта, водитель и его троллейбус...

У каждой парты может быть только один студент. И студенту принадлежит только одна парта.

На уровне БД просто создается поле в котором хранится `id` связанного объекта с признаком уникальности.



models - relationships

M-to-M

Многие ко многим

Поле `ManyToManyField`

Одной записи в таблице принадлежит много записей в таблице другого типа, но и другим записям в первой таблице могут принадлежать эти же объекты: студент и его курсы, пост и теги.

Каждый студент может выбрать себе много курсов и эти же самые курсы может выбрать другой студент.

На уровне БД создается дополнительная таблица привязки в которой есть два поля с `id` одного объекта и с `id` второго объекта.



Дока наше все!

[https://docs.djangoproject.com/en/1.7/
topics/db/models/](https://docs.djangoproject.com/en/1.7/topics/db/models/)

[https://docs.djangoproject.com/en/1.7/
ref/models/fields/](https://docs.djangoproject.com/en/1.7/ref/models/fields/)



QuerySets

Созданы для общения с БД

Создание записей в БД

Выборка данных из БД

Редактирование данных в БД

Удаление записей



QuerySets

Создание записей в БД

```
john = Student()  
john.name = 'John'  
john.save()
```

```
john = Student(name = 'John')  
john.save()
```

```
john = Student.objects.create(name='John')
```



QuerySets

Выборка данных из БД

```
student = Student.objects.get(id=1)
```

```
student = Student.objects.get(name='John')
```

```
students = Student.objects.all()
```

```
students = Student.objects.filter(package='vip')
```

```
students = Student.objects.filter(id__in=[1,2,3])
```

```
students = Student.objects.filter(name__startswith='vip')
```

```
student = Student.objects.all()[0]
```



QuerySets

Редактирование данных в БД

```
student = Student.objects.get(id=1)  
student.name='John'  
student.save()
```



QuerySets

Удаление записей

```
student = Student.objects.get(id=1)  
student.delete()
```



Дока наше все!

[https://docs.djangoproject.com/en/1.7/
topics/db/queries/](https://docs.djangoproject.com/en/1.7/topics/db/queries/)

[https://docs.djangoproject.com/en/1.7/
ref/models/querysets/](https://docs.djangoproject.com/en/1.7/ref/models/querysets/)



Models to admin

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

Админка автоматически строит интерфейс работы с данными по зарегистрированным моделям. При этом для каждого поля строится виджет исходя из типа поля прописанного в модели.

