

Python

базовый тренинг

Ружин Алексей
ruzin@me.com

Django - ORM

- ```
class Item(models.Model):
 text = models.CharField()
 number = models.IntegerField()
 real = models.FloatField()
 dec = models.DecimalField()
 dt = models.DateField()
 dt_time = models.DateTimeField()
 fk = models.ForeignKey('Model', on_delete=...)
```

# Django - создание

- `item = Item()`
- `item = Item(text='asdf', number=23,...)`
- `item.save()`
- `item = Item.objects.create(  
 text='asdf',  
 number=23,  
 ...  
)`
- `item, created = Item.objects.get_or_create(  
 text='asdf',  
 defaults={'number': 23, ...}  
)`

# Обновление и удаление

- `item.text = 'xyz'`  
`item.save()`  
`item.save(update_fields=['text'])`
- `Item.objects.filter(...).update(number=123)`
- `item.delete()`
- `Item.objects.filter(...).delete()`

# Django - Выборка

- `Item.objects.get(pk=123)`
- `Item.objects.filter(text='abc')`
- `Item.objects.exclude(text='abc')`
- `Item.objects.filter(text='abc').exclude(number__gte=12).order_by('-text')`
- `for item in Item.objects.filter(...):  
 print(item.text)`
- `Item.objects.filter(...).exists()` # True/False  
`Item.objects.filter(...).count()` # кол-во подходящих моделей  
`Item.objects.filter(...)[:10]` # получить первые 10 моделей

# Выборка части полей

- `Item.objects.values('text').distinct()`  
`[{'text': '...'}, {'text': '...'}, ...]`
- `Item.objects.values_list('text', flat=True).distinct()`  
`['abc', 'xyz', ...]`

# Агрегатные функции

- `Item.objects.values('number').annotate(  
 cnt=Count('pk'),  
 max_pub_date=Max('pub_date'),  
 min_pub_date=Min('pub_date')  
)`  
`[{'number': ..., 'cnt': ..., 'max_pub_date': ...}, ...]`
- `Item.objects.aggregate(cnt=Count('pk'))`  
`{'cnt': ...}`

# Q, F

- text='abc' or text='xyz' ?  
Item.objects.filter(Q(text='abc') | Q(text='xyz'))
- q1 = Q(text='abc')  
q2 = Q(...)  
q3 = Q(...)  
q = Q()  
q &= q1 | q2  
q &= q3  
Item.objects.filter(q)
- field1 > field2 ?  
Item.objects.filter(field1\_\_gt=F('field2'))



# СВЯЗАННЫЕ МОДЕЛИ

- ```
class Category(models.Model):  
    pass
```
- ```
class Item(models.Model):
 category = models.ForeignKey(
 Category,
 related_name='items',
 on_delete=models.CASCADE
)
```
- ```
item = Item.objects.get(pk=1)  
item.category.__class__ == Category
```
- ```
category = Category.objects.get(pk=1)
category.items.__class__ == QuerySet
for item in category.items.all():
 print(item)
```

# МНОГИЕ КО МНОГИМ

- ```
class Book(models.Model):  
    title = models.CharField()  
    authors = models.ManyToManyField(  
        'Author',  
        related_name='books'  
    )
```
- ```
class Author(models.Model):
 name = models.CharField()
```
- ```
author.books.add(book)  
book.authors.add(author)
```
- ```
author.books.all()
book.authors.all()
```

# Сложные запросы

- Найти всех соавторов для Olga:

```
Author.objects.filter(
 ~Q(name='Olga'),
 books__author__name='Olga'
)
.distinct()
```

# Оптимизация выборки связанных моделей

- `class Publisher(models.Model):`  
    `title = CharField(...)`
- `class Book(models.Model):`  
    `...`  
    `publisher = models.ForeignKey(Publisher)`
- `Book.objects.filter(...).select_related('publisher')`
- `Author.objects.filter(...).prefetch_related('books')`

# Тестирование

- `shop/tests.py`
- `manage.py test shop`

# Тесты

- `class MyTests(TestCase):`  
  
    `def test_models(self):`  
        `self.assertEqual(2, 2)`  
  
    `def test_views(self):`  
        `response = self.client.get(reverse('shop:index'))`  
        `self.assertEqual(response.status_code, 200)`

# uwsgi

- `pip install uwsgi`
- `uwsgi.ini`
- `uwsgi —ini uwsgi.ini`

# uwsgi.ini

- `[uwsgi]`  
chdir=/Users/ruzin/Projects/demo-django/shop/  
module=app.wsgi  
home=/Users/ruzin/.virtualenvs/demodjango  
  
master=True  
processes=10  
vacuum=True  
max-requests=5000  
http=127.0.0.1:8000  
check-static=/Users/ruzin/Projects/demo-django/shop/