

#9. Создание связей между моделями через класс ForeignKey

[Django для python \(уроки\)](#)

[youtube](#)

[propoprogs.ru](#)

[github](#)

[Русскоязычная документация по Django 3](#)

Связывание таблиц

Разделение данных между таблицами

нормализация таблиц

добавляем поле ключ индентификатор на таблицу категории

добавим еще одно поле cat_id в таблицу women внешний ключ

category определим два поля: идентификатор id и название раздела – name

три класса организации данных

- ForeignKey – для связей Many to One (поля отношений);
- ManyToManyField – для связей Many to Many (многие ко многим);
- OneToOneField – для связей One to One (один к одному).

Связи

классом ForeignKey организации связей «многие к одному» или «одного ко многим» Many to One

Класс ForeignKey параметры

- ссылка или строка класса модели
- опция on_delete

on_delete значения:

- models.CASCADE – при удалении записи из первичной модели происходит удаление всех записей из вторичной модели, связанных с удаляемой категорией;
- models.PROTECT – запрещает удаление записи из первичной модели, если она используется во вторичной (выдает исключение);
- models.SET_NULL – при удалении записи первичной модели устанавливает значение foreign key в NULL у соответствующих записей вторичной модели;
- models.SET_DEFAULT – то же самое, что и SET_NULL, только вместо значения NULL устанавливает значение по умолчанию, которое должно быть определено через класс ForeignKey;
- models.SET() – то же самое, только устанавливает пользовательское значение;
- models.DO_NOTHING – удаление записи в первичной модели не вызывает никаких действий у вторичных моделей.

women/models.py

модель для категории

```
class Category(models.Model):
    name = models.CharField(max_length=100, db_index=True)
    # db_index индексировать данное поле (быстрый поиск)
    # два индексируемых поля: id и name

    def __str__(self):
        return self.name
```

поле cat_id во вторичной модели Women

```
cat = models.ForeignKey('Category', on_delete = models.PROTECT, null=True)
# _id автоматически
# функцию PROTECT запрещает удаление если используется
# null=True разрешить пустые поля
```

формируем новую миграцию

```
python manage.py makemigrations
```

внести изменения в бд

```
python manage.py migrate
```

перейти в режим шела для работы с моделями

```
python manage.py shell
```

```
# импортируем модели
from women.models import *

# две записи в таблице

Category.objects.create(name='Актрисы')
Category.objects.create(name='Певицы')
# у всех записей таблицы women установим поле cat_id равным 1
w_list = women.objects.all() # выбрать все записи

w_list.update(cat_id=1) # обновить

from django.db import connection
connection.queries # показать созданные SQL запросы
```

Классы моделей и их экземпляры

```
women.title
w1 = women(title='t1', content='c1', cat_id=1)
type(w1.title)
type(w1.cat)
print(w1.id, w1.time_create, w1.time_update)
from django.db import connection
connection.queries
w1.cat.name
w2 = women.objects.get(pk=2)
w2.cat
```

Отображение категорий в шаблоне

в шаблоне base.html

```
{% if cat_selected == 0 %}
    <li class="selected">Все категории</li>
{% else %}
    <li><a href="{% url 'home' %}">Все категории</a></li>
{% endif %}

{% for c in cats %}
    {% if c.pk == cat_selected %}
        <li class="selected">{{c.name}}</li>
    {% else %}
        <li><a href="{{ c.get_absolute_url }}">{{c.name}}</a></li>
    {% endif %}
{% endfor %}
```

в модели Category

```
def get_absolute_url(self):
    return reverse('category', kwargs={'cat_id': self.pk})
```

в файле women/urls.py

```
path('category/<int:cat_id>/', show_category, name='category'),
```

в файле women/views.py

```
def show_category(request, cat_id):
    return HttpResponse(f"Отображение категории с id = {cat_id}")

def index(request):
    posts = women.objects.all()
    cats = Category.objects.all()

    context = {
        'posts': posts,
        'cats': cats,
        'menu': menu,
        'title': 'Главная страница',
        'cat_selected': 0,
    }
```

```
return render(request, 'women/index.html', context=context)
```

в функцию show_category

```
def show_category(request, cat_id):
    posts = women.objects.filter(cat_id=cat_id)
    if len(posts) == 0:
        raise Http404()
    cats = Category.objects.all()

    context = {
        'posts': posts,
        'cats': cats,
        'menu': menu,
        'title': 'Главная страница',
        'cat_selected': cat_id,
    }

    return render(request, 'women/index.html', context=context)
```

В шаблоне index.html

```
<li><div class="article-panel">
    <p class="first">Категория: {{p.cat.name}}</p>
    <p class="last">Дата: {{p.time_update|date:"d-m-Y H:i:s"}}</p>
</div>
```