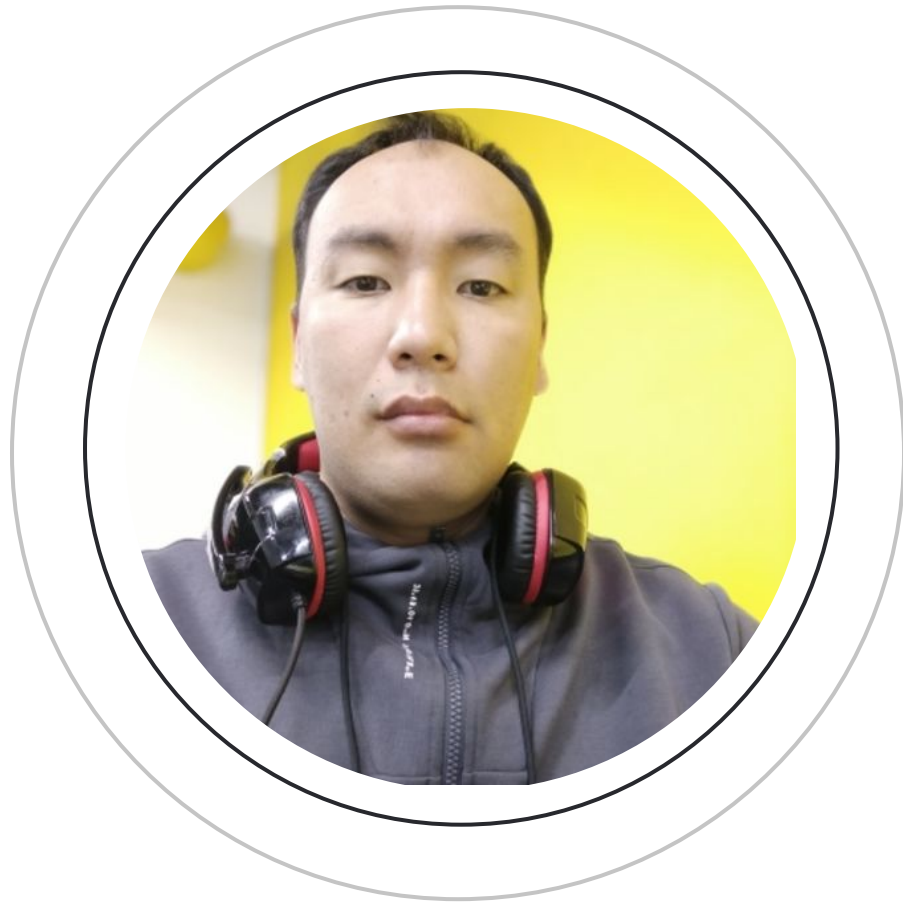


Разбор вопросов



Адилет Асанкожоев



План занятия

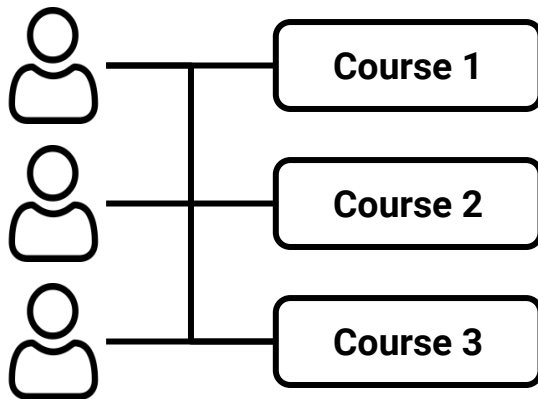
- 1 [Связи многие ко многим в Django](#)
- 2 [M2M с промежуточной моделью](#)
- 3 [Продвинутая админка для M2M](#)
- 4 [Параметризация запросов](#)

Связи “многие ко многим” в Django



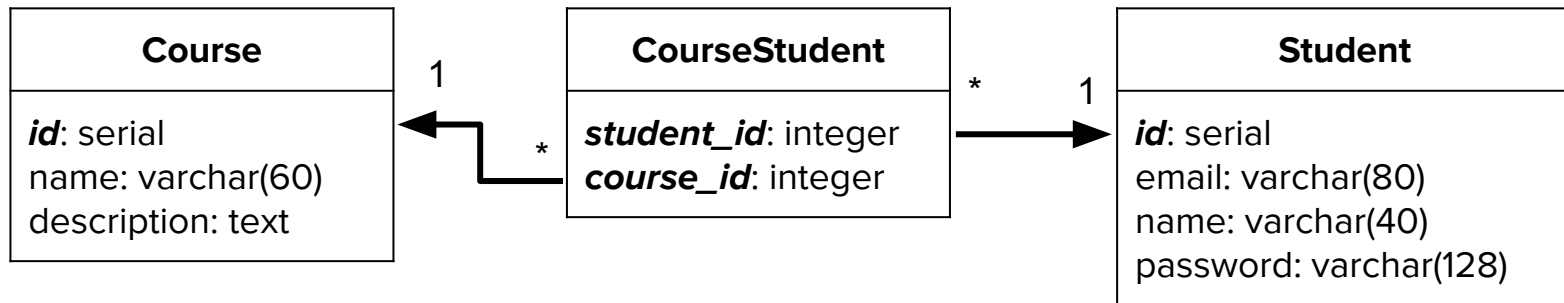
1

Связи “многие ко многим” (m2m)



Связь реализуется с помощью атрибута `models.ManyToManyField`

Классический пример M2M



```
class Student(models.Model):  
    email = models.CharField(max_length=80)  
    name = models.CharField(max_length=40)  
    password = models.CharField(max_length=128)
```

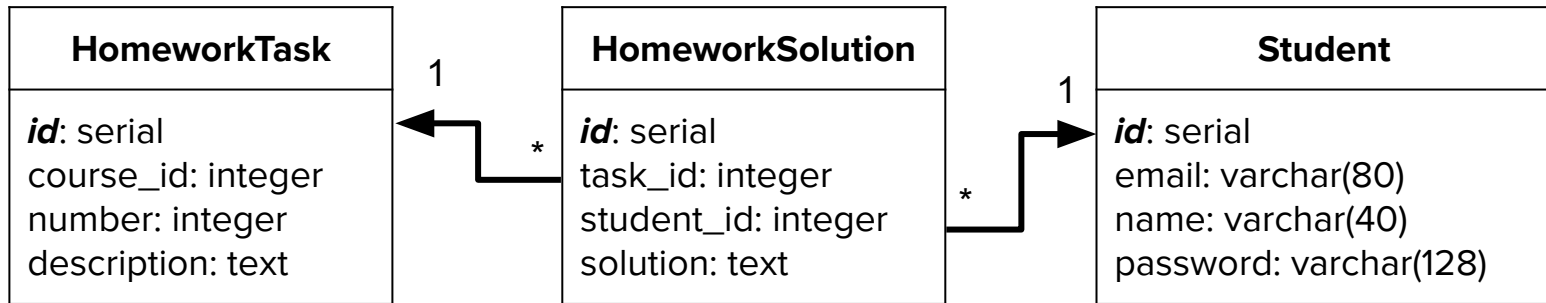
```
class Course(models.Model):  
    name = models.CharField(max_length=60)  
    description = models.TextField()  
    students = models.ManyToManyField(Student, related_name='courses')
```

M2M с промежуточной моделью



2

through



```
class Student(models.Model):  
    ...  
  
class HomeworkTask(models.Model):  
    number = models.IntegerField()  
    description = models.TextField()  
    students = models.ManyToManyField(Student, related_name='courses', through='HomeworkSolution')  
  
class HomeworkSolution(models.Model):  
    task = models.ForeignKey(HomeworkTask, on_delete=models.CASCADE, related_name='solutions')  
    student = models.ForeignKey(Student, on_delete=models.CASCADE, related_name='solutions')  
    solution = models.TextField()
```


Продвинутая админка для M2M



3

Советы

- Делаем `__str__` в моделях
- Указываем настройки отображения в админ-классе: `list_display`, `list_filter` и т.п.

<https://docs.djangoproject.com/en/3.2/ref/contrib/admin/#modeladmin-options>

- `Inlines` — для встраивания отображения таблицы в другой таблице при создании/обновлении строк.
- `InlineFormset` — правила работы и отображения встроенных таблиц.

<https://docs.djangoproject.com/en/3.2/ref/contrib/admin/#inlinemodeladmin-objects>

Параметризация запросов



4

Необязательные параметры

Передаются **после** символа **?** в конце запроса:

```
https://example.org?name=ivan
```

В Django их можно получить из **объекта** **request**. Например:

```
request.GET.get('name')
```

Обязательные параметры

```
from django.urls import path
from . import views

urlpatterns = [
    path('hello/<name>/', views.hello),
    path('articles/<int:year>/<int:month>/<int:day>/', views.article_detail),
    path('articles/<int:year>-<int:month>-<int:day>/', views.article_detail),
]

def hello(request, name):
    response = f'Здравствуйте, {name}!'
    return HttpResponse(response)

def article_detail(request, year, month, day):
    response = f'Статья за {year} год {month} месяц {day} день!'
    return HttpResponse(response)
```

**Задавайте вопросы
и пишите отзыв о лекции**

