QuerySets – ленивы, создание [QuerySet](https://djbook.ru/rel3.0/ref/models/querysets.html#django.db.models.query.QuerySet) не выполняет запросов к базе данных. Вы можете добавлять фильтры хоть весь день и Django не выполнит ни один запрос, пока [QuerySet](https://djbook.ru/rel3.0/ref/models/querysets.html#django.db.models.query.QuerySet) не вычислен

get()

filter()

all()

exclude()

model\_name()

save()

delete()

[::]

Further filtering or ordering of a sliced queryset is prohibited due to the ambiguous nature of how that might work.

Отрицательные индексы (например, Entry.objects.all()[-1]) не поддерживаются.

Django предоставляет [класс F](https://djbook.ru/rel3.0/ref/models/expressions.html#django.db.models.F) для таких сравнений. Экземпляр F() рассматривается как ссылка на другое поле модели. Эти ссылки могут быть использованы для сравнения значений двух разных полей одного объекта модели.

>>> **from** django.db.models **import** F

>>> Entry**.**objects**.**filter(number\_of\_comments\_\_gt**=**F('number\_of\_pingbacks'))

Вы можете использовать два нижних подчеркивания для использования полей связанных объектов в F(). Объект F() с двойным нижним подчеркиванием обеспечит все необходимые JOIN для получения необходимых связанных объектов. Например, чтобы получить все записи, у которых имя автора совпадает с названием блога, нужно выполнить такой запрос:

>>> Entry**.**objects**.**filter(authors\_\_name**=**F('blog\_\_name'))

**Caching and**QuerySet**s**

Manager.raw()

Если вам необходимо выполнить более сложные вопросы (например, запросы с OR), вы можете использовать [объект Q](https://djbook.ru/rel3.0/ref/models/querysets.html#django.db.models.Q).