


1. Получение всех объектов:

Чтобы получить все товары:

python

 Копировать код


```
products = Product.objects.all()
```

Это вернет все записи из таблицы `Product`.

2. Фильтрация объектов:

Метод `filter()` позволяет получить записи, соответствующие условиям:

python


 Копировать код

```
electronics = Product.objects.filter(category__name="Электроника")
```

3. Получение одного объекта:

Если тебе нужен конкретный объект, можно использовать `get()`:

python

 Копировать код

```
iphone = Product.objects.get(name="iPhone 14")
```

Важно: `get()` возвращает только один объект. Если объектов будет больше одного или не будет совсем, вызовется ошибка.

4. Порядок сортировки:

Чтобы отсортировать объекты по цене, можно использовать `order_by()`:

python


 Копировать код

```
products_by_price = Product.objects.order_by('price') # По возрастанию  
products_by_price_desc = Product.objects.order_by('-price') # По убыванию
```

5. Ограничение выборки:

С помощью `limit` (через срезы) можно ограничить количество записей:

python


 Копировать код

```
top_5_products = Product.objects.all()[:5] # Первые 5 товаров
```

6. Аннотации и агрегирование:

Если нужно посчитать количество товаров в категории:

python

 Копировать код


```
from django.db.models import Count

categories_with_count = ProductCategory.objects.annotate(num_products=Count('product'))
for category in categories_with_count:
    print(f'{category.name}: {category.num_products} товаров')
```

7. Использование `values()` и `values_list()`:

Эти методы позволяют извлекать только определенные поля:

python


 Копировать код

```
product_names = Product.objects.values_list('name', flat=True) # Список названий товаров
```

8. Поиск по частичному совпадению:

Метод `filter()` поддерживает операции для частичного поиска. Например, поиск товаров, название которых начинается с "i":

python


 Копировать код

```
products_with_i = Product.objects.filter(name__startswith="i")
```

9. Удаление объектов:

Чтобы удалить объекты, можно использовать метод `delete()`:

python

 Копировать код


```
old_products = Product.objects.filter(quantity=0)
old_products.delete()
```



10. Обновление записей:

Для массового обновления используется метод `update()` :

python


 Копировать код

```
Product.objects.filter(category__name="Скидки").update(price=99.99)
```

15. Фильтрация по диапазонам:

Для работы с диапазонами можно использовать `__range` :

python


 Копировать код

```
products_in_price_range = Product.objects.filter(price__range=(500, 1500))
```

17. Подсчет количества объектов:

Чтобы узнать количество товаров, используй метод `count()` :

python

 Копировать код

```
total_products = Product.objects.count()
```

- `union()` возвращает объединенные результаты двух запросов:

python

Копировать код

```
query1 = Product.objects.filter(price__lte=500)
query2 = Product.objects.filter(price__gte=1500)
combined = query1.union(query2)
```

- `intersection()` возвращает общие результаты двух запросов:

python

Копировать код

```
query1 = Product.objects.filter(price__gte=500)
query2 = Product.objects.filter(quantity__gte=1)
intersected = query1.intersection(query2)
```

- `difference()` возвращает разницу между результатами двух запросов:

python

Копировать код

```
query1 = Product.objects.filter(price__gte=500)
query2 = Product.objects.filter(price__gte=1000)
difference = query1.difference(query2)
```

19. Лимит и смещение (`limit` и `offset`):

Django поддерживает срезы для ограничения количества результатов и сдвига. Это полезно для пагинации:

python

Копировать код

```
products_page_1 = Product.objects.all()[:10] # первые 10 товаров
products_page_2 = Product.objects.all()[10:20] # следующие 10 товаров
```

20. Фильтрация с `exclude()`:

Метод `exclude()` позволяет исключить объекты, соответствующие условиям:

python

Копировать код

```
non_discounted_products = Product.objects.exclude(category__name="Скидки")
```