

Практическая работа №3.1

Задание №1

Напишите запрос на создание 6-7 новых автовладельцев и 5-6 автомобилей, каждому автовладельцу назначьте удостоверение и от 1 до 3 автомобилей. Задание можете выполнить либо в интерактивном режиме интерпретатора, либо в отдельном python-файле. Результатом должны стать запросы и отображение созданных объектов.

```
>>> from practical_work_3_app.models import Car
>>> Car(id=0, state_number="0", brand="Tesla", model="S", color="red").save()
>>> Car(id=1, state_number="1", brand="Tesla", model="S", color="red").save()
>>> Car(id=2, state_number="2", brand="Tesla", model="S", color="red").save()
>>> Car(id=3, state_number="3", brand="Tesla", model="S", color="red").save()
>>> Car(id=4, state_number="4", brand="Tesla", model="S", color="red").save()
>>> Car(id=5, state_number="5", brand="Tesla", model="S", color="red").save()

>>> from practical_work_3_app.models import CarOwner
>>> CarOwner(id=0, first_name="CarOwner", last_name="0", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=1, first_name="CarOwner", last_name="1", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=2, first_name="CarOwner", last_name="2", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=3, first_name="CarOwner", last_name="3", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=4, first_name="CarOwner", last_name="4", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=5, first_name="CarOwner", last_name="5", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()
>>> CarOwner(id=6, first_name="CarOwner", last_name="6", birth_day="2023-03-12", passport_number="123456", address="Test address", nationality="RU").save()

>>> for i in range(6):
...     Property(id=i, owner_id=CarOwner.objects.get(id=i), car_id=Car.objects.get(id=i), start_date="2023-03-12", end_date="2023-03-14").save()
>>>
```

Задание №2

По созданным в пр.1 данным написать следующие запросы на фильтрацию:

- Выведете все машины марки "Tesla"
- Найти всех водителей с именем "CarOwner"
- Взяв любого случайного владельца получить его id, и по этому id получить экземпляр удостоверения в виде объекта модели
- Вывести всех владельцев красных машин
- Найти всех владельцев, чей год владения машиной начинается с 2023

```
>>> Car.objects.filter(brand="Tesla")
<QuerySet [ <Car: Car object (0)>, <Car: Car object (1)>, <Car: Car object (2)>, <Car: Car object (3)>, <Car: Car object (4)>, <Car: Car object (5)>]>

>>> CarOwner.objects.filter(first_name="CarOwner")
<QuerySet [ <CarOwner: CarOwner object (0)>, <CarOwner: CarOwner object (1)>, <CarOwner: CarOwner object (2)>, <CarOwner: CarOwner object (3)>, <CarOwner: CarOwner object (4)>, <CarOwner: CarOwner object (5)>, <CarOwner: CarOwner object (6)>]>

>>> DriversLicense.objects.get(owner_id=CarOwner.objects.get(id=0))
<DriversLicense: DriversLicense object (0)>

>>> CarOwner.objects.filter(property__car_id__color="red")
<QuerySet [ <CarOwner: CarOwner object (0)>, <CarOwner: CarOwner object (1)>, <CarOwner: CarOwner object (2)>, <CarOwner: CarOwner object (3)>, <CarOwner: CarOwner object (4)>, <CarOwner: CarOwner object (5)>]>

>>> CarOwner.objects.filter(driverslicense__date_of_license__gte="2023-01-01")
<QuerySet [ <CarOwner: CarOwner object (0)>, <CarOwner: CarOwner object (1)>, <CarOwner: CarOwner object (2)>, <CarOwner: CarOwner object (3)>, <CarOwner: CarOwner object (4)>, <CarOwner: CarOwner object (5)>, <CarOwner: CarOwner object (6)>]>
```

Задание №3

Необходимо реализовать следующие запросы:

- Вывод даты выдачи самого старшего водительского удостоверения

- Укажите самую позднюю дату владения машиной, имеющую какую-то из существующих моделей в вашей базе
- Выведите количество машин для каждого водителя
- Подсчитайте количество машин каждой марки
- Отсортируйте всех автовладельцев по дате выдачи удостоверения

```
[>>> DriversLicense.objects.aggregate(Min('date_of_license'))
{'date_of_license__min': datetime.date(2023, 3, 12)}
>>>
```

```
>>> from django.db.models import Max
>>> DriversLicense.objects.aggregate(Max('date_of_license'))
{'date_of_license__max': datetime.date(2023, 3, 12)}
>>>
```

```
[>>> Property.objects.values('owner_id_id').annotate(Count('car_id'))
<QuerySet [{ 'owner_id_id': 0, 'car_id_count': 1}, { 'owner_id_id': 1, 'car_id_count': 1}, { 'owner_id_id': 2, 'car_id_count': 1}, { 'owner_id_id': 3, 'car_id_count': 1}, { 'owner_id_id': 4, 'car_id_count': 1}, { 'owner_id_id': 5, 'car_id_count': 1}]>
>>>
```

```
[>>> Car.objects.values('brand').annotate(Count('brand'))
<QuerySet [{ 'brand': 'Tesla', 'brand_count': 6}]>
>>>
```

```
[>>> DriversLicense.objects.values('owner_id_id').order_by('date_of_license')
<QuerySet [{ 'owner_id_id': 0}, { 'owner_id_id': 1}, { 'owner_id_id': 2}, { 'owner_id_id': 3}, { 'owner_id_id': 4}, { 'owner_id_id': 5}, { 'owner_id_id': 6}]>
>>>
```