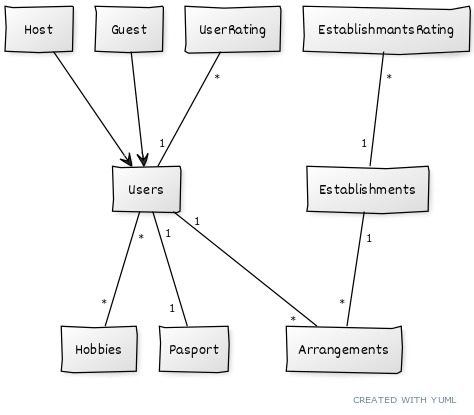
Задания по занятию 4.

1. Создать диаграмму зависимости моделей:



1. Создать модели и связать их внешними ключами – **сделано**
2. Проработать логику проекта, добавив необходимые ограничения – **сделано**
3. Зарегистрировать модели в admin.py – **сделано**
4. Добавить 5-6 объектов для каждой модели в проекте – **сделано**
5. Вывести всех пользователей старше 30 которые любят спорт

users=Users.objects.filter(age\_\_gt=30,hobbies\_\_category="sp").distinct()  
 users

<QuerySet [<Users: Nataly>]>

1. Вывести первых 5 девушек у которых больше 3 хобби

girls=Users.objects.values('name').annotate(count\_hobby=Count('hobbies\_\_name')).filter(count\_hobby\_\_gte=3,sex='f').order\_by('-count\_hobby')[:5]

girls

<QuerySet [{'name': 'Valeria', 'count\_hobby': 4}, {'name': 'Aryna', 'count\_hobby': 3}, {'name': 'Alesia', 'count\_hobby': 3}, {'name': 'Nataly', 'count\_hobby': 3}]>

1. Вывести все заведения с максимальным рейтингом:

establishments = Establishments.objects.filter(establishmentsrating\_\_rating="5").distinct()  
establishments

<bound method QuerySet.distinct of <QuerySet [<Establishments: Umami>, <Establishments: Therry>]>>

1. Подключить PostgreSQL - **сделано**

Задания по занятию 5:

1. Создать наследование моделей и сделать миграции – **сделано**
2. Установить Django-debug-toolbar – **сделано**
3. Создать индексы для Host и Guest по которым чаще всего осуществляется поиск: имя, бюджет, возраст, хобби.
4. Создать индексы для Establishments

Задания по занятию 6:

1.

2. users=Users.objects.aggregate(age\_avg=Avg('age'))

users

users=Users.objects.filter(age\_\_gte=29.3)

3. users=Users.objects.filter(age\_\_gt=30,hobbies\_\_category="sp").distinct()

4. girls=Users.objects.filter(sex='f',age\_\_range=(18,30)) (с рейтингом не получилось вывести, так как почему то указывая Userrating мне среднее давало не самого рейтинга, а среднее число пользователей в классе Userrating).

5.girls=Users.objects.values('name').annotate(count\_hobby=Count('hobbies\_\_name')).filter(count\_hobby\_\_gte=3,sex='f').order\_by('-count\_hobby')[:5]

6.

7. users=Users.objects.annotate(num\_hobbies=Count('hobbies')).filter(num\_hobbies\_\_gt=5)  
 users