**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**
**(БГТУ им. В.Г.Шухова)**

Лабораторная работа №4
дисциплина «Технологии web-программирования»
по теме «Разработка и проектирование базы данных»

Выполнил: студент группы ВТ-41 Макаров Д.С.
Проверил: Картамышев С.В.

Белгород 2020

# Лабораторная работа №4

## «Разработка и проектирование базы данных»

**Цель работы:**изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

### Задание к лабораторной работе:

1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных (ORM, Active Record).
3. Разработать структуру базы данных.
4. Разработать соответствующие модели в приложении.
5. В отчёт приложить схему базы данных, а так же код одной из моделей (на своё усмотрение).

### Ход работы

В качестве СУБД был использовал PostgreSQL развернутый в Docker контейнере. Была разработана следующая структура базы данных.

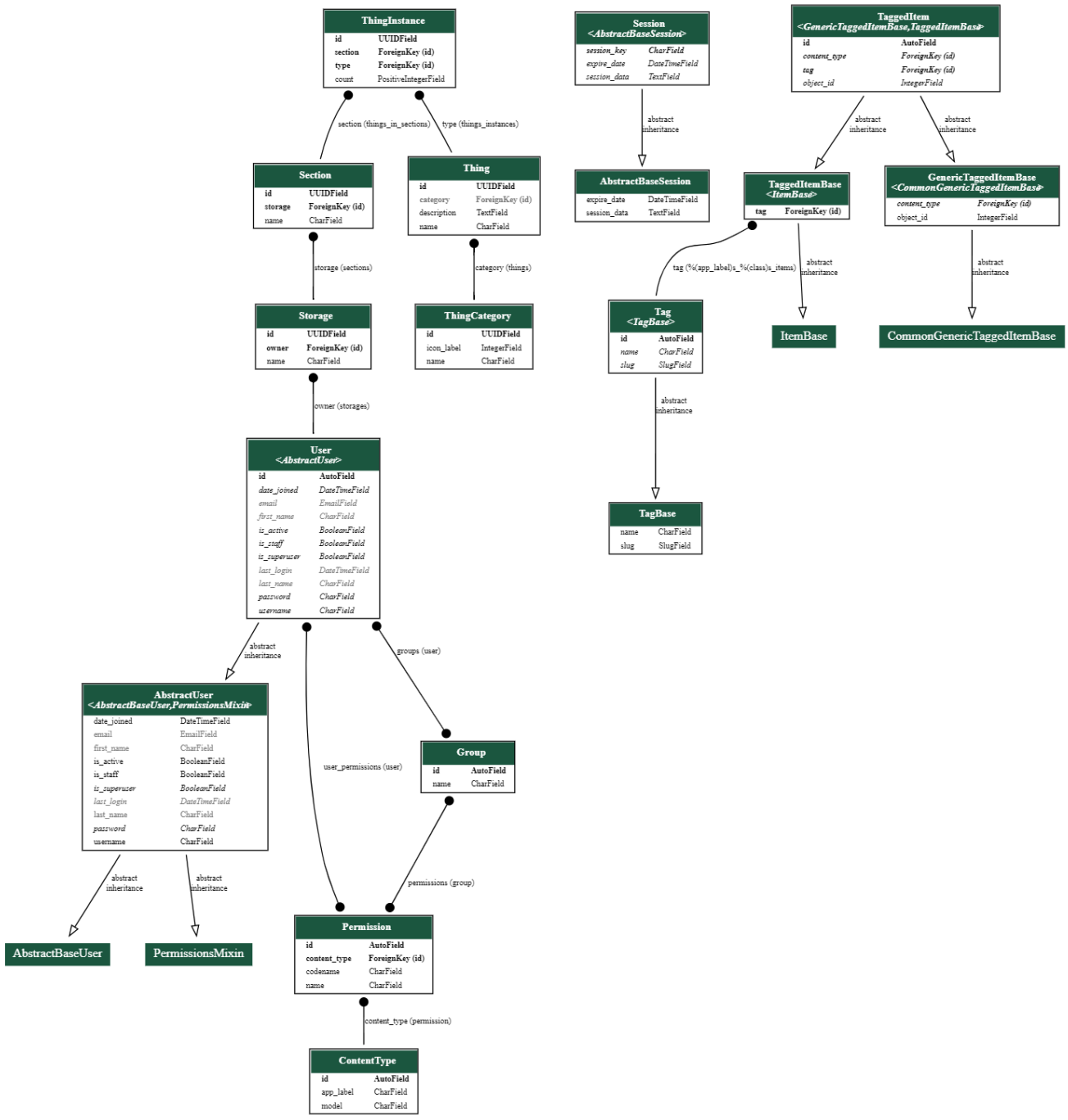


Рис. 1: Схема базы данных

Таблицы *User*, *AbstractUser*, *Group*, *Permission*, *Session* были предоставлены ORM в составе фреймворка Django.

Все таблицы связанные с тэгами предоставлены библиотекой *django-taggit*.

Таблицы Section, Storage, Thing, ThingInstance, ThingCategory был описаны в моделях (см. в приложении).

# Приложение

## Содержимое файла storagesModels.py

```
from uuid import uuid4
from django.db import models
from taggit.managers import TaggableManager

class Storage(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid4, editable=False)
    name = models.CharField(max_length=128)
    owner = models.ForeignKey('auth.User',related_name='storages',on_delete=models.CASCADE)
    tags = TaggableManager()
    def __str__(self):
        return str(id)

class Section(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid4, editable=False)
    name = models.CharField(max_length=128)
    storage = models.ForeignKey(
        'Storage',
        related_name='sections',
        on_delete=models.CASCADE
    )

    def __str__(self):
        return str(id)
```

## Содержимое файла thingsModels.py

```
from uuid import uuid4
from django.db import models
from taggit.managers import TaggableManager

class Thing(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid4, editable=False)
    name = models.CharField(max_length=100)
    description = models.TextField()
    category = models.ForeignKey(
        'ThingCategory',
        related_name='things',
        blank=True,
        null=True,
        on_delete=models.CASCADE
    )

    tags = TaggableManager()
    def __str__(self):
        return str(id)

class ThingInstance(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid4, editable=False)
    section = models.ForeignKey(
        'storages.Section',
        related_name='things_in_sections',
        on_delete=models.CASCADE
    )
    type = models.ForeignKey(
        'Thing',
        related_name='things_instances',
```

```
        on_delete=models.CASCADE
    )
    count = models.PositiveIntegerField()

    def __str__(self):
        return str(id)

class ThingCategory(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid4, editable=False)
    name = models.CharField(max_length=100)
    icon_label = models.IntegerField()

    def __str__(self):
        return str(id)
```