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

Лабораторная работа по теме

«Разработка и проектирование базы данных»  
дисциплина:

Технологии web-программирования

Выполнил: ст. группы ВТ-41   
Фаракшин Н. Р.  
Проверил: ст. пр. 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).

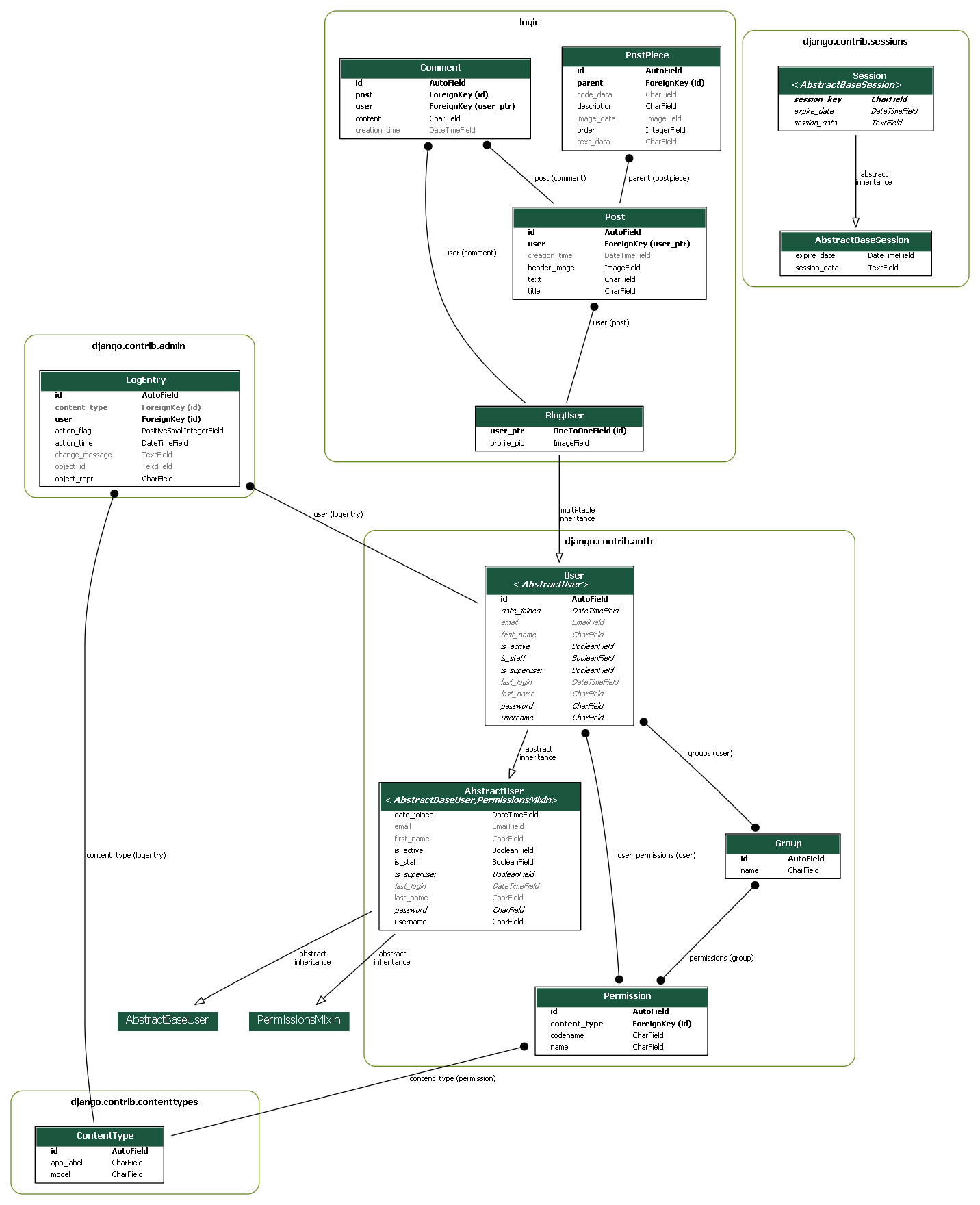
**Выполнение работы**

В качестве основы для разработки базы данных была выбрана СУБД PostgreSQL. Взаимодействие с базой данных будет осуществляться с помощью встроенного в фреймворк Django ORM. Структура базы данных должна содержать такие сущности, как:

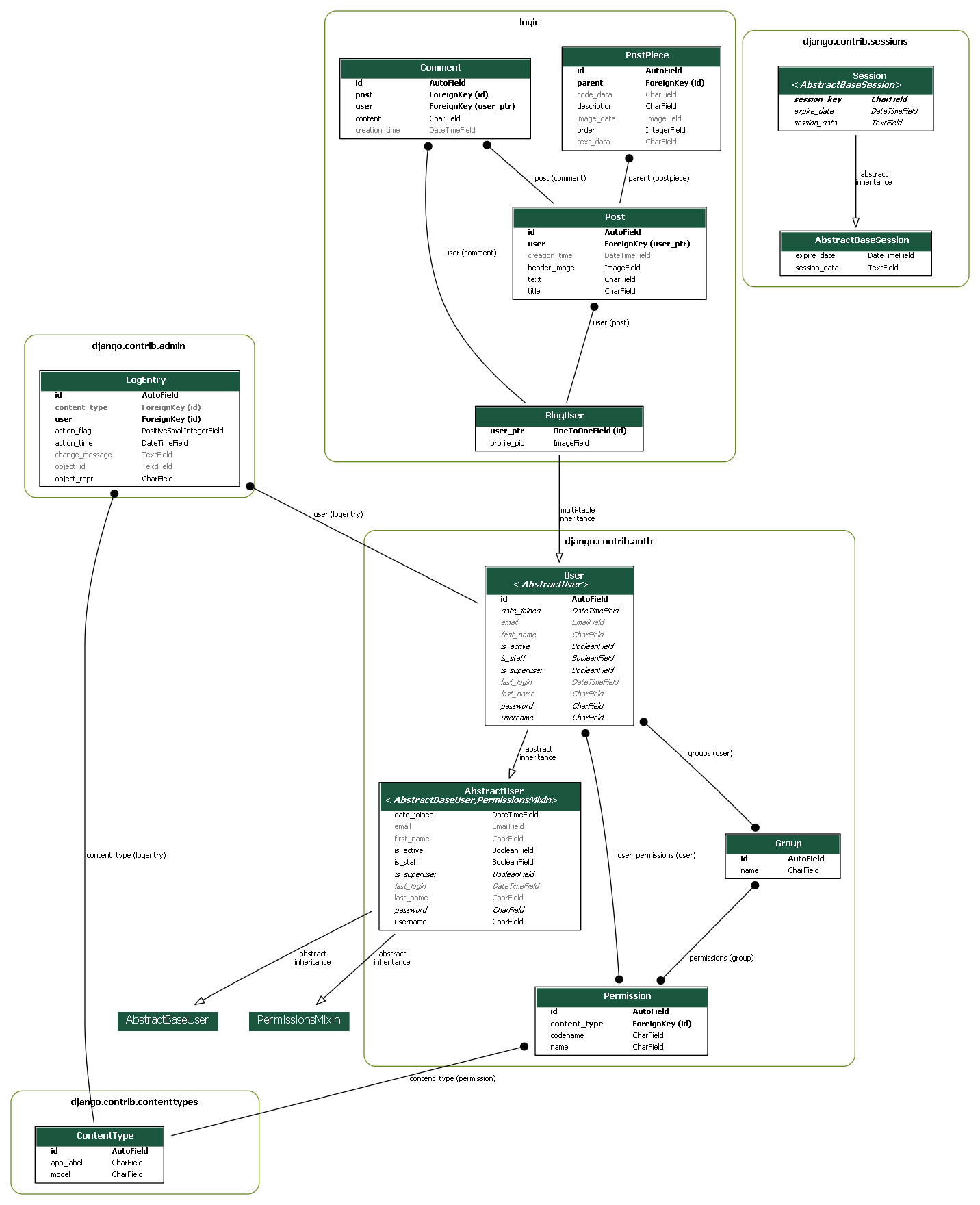
* Пользователь
* Пост
* Комментарий.

Модель пользователя сделаем на основе стандартной модели пользователя Django, добавив к ней поле «Фото в профиле».

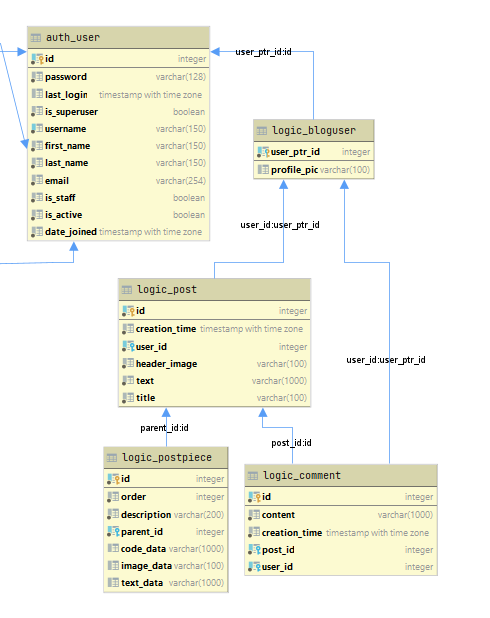
Структура базы данных, включающая в себя также стандартные модели Django:



Исключив встроенные модели, получим следующую структуру таблиц:



Листинги файлов с описаниями моделей приведены в приложении.   
Структура таблиц базы данных, полученной после применения миграций, следующая:



**Приложение**

Исходный код файла models.py

from django.contrib.auth.models import User

from django.db import models

class BlogUser(User):

    profile\_pic = models.ImageField(upload\_to="media/profile\_images", null=True)

class Post(models.Model):

    user = models.ForeignKey(BlogUser, on\_delete=models.CASCADE)

    title = models.CharField(max\_length=100)

    text = models.CharField(max\_length=1000)

    header\_image = models.ImageField(upload\_to="media/posts\_images")

    creation\_time = models.DateTimeField(auto\_now\_add=True)

class PostPiece(models.Model):

    order = models.IntegerField()

    parent = models.ForeignKey(Post, on\_delete=models.CASCADE)

    description = models.CharField(max\_length=200)

    image\_data = models.ImageField(upload\_to="media/posts\_images", null=True, blank=True)

    text\_data = models.CharField(max\_length=1000, null=True, blank=True)

    code\_data = models.CharField(max\_length=1000, null=True, blank=True)

class Comment(models.Model):

    user = models.ForeignKey(BlogUser, on\_delete=models.CASCADE)

    post = models.ForeignKey(Post, on\_delete=models.CASCADE)

    content = models.CharField(max\_length=1000)

    creation\_time = models.DateTimeField(auto\_now\_add=True)