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»

тема: «Разработка и проектирование базы данных»

Выполнил: студент группы ВТ-41

Ковалёв И. Д.

Проверил:

Картамышев С. В.

Белгород 2020

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

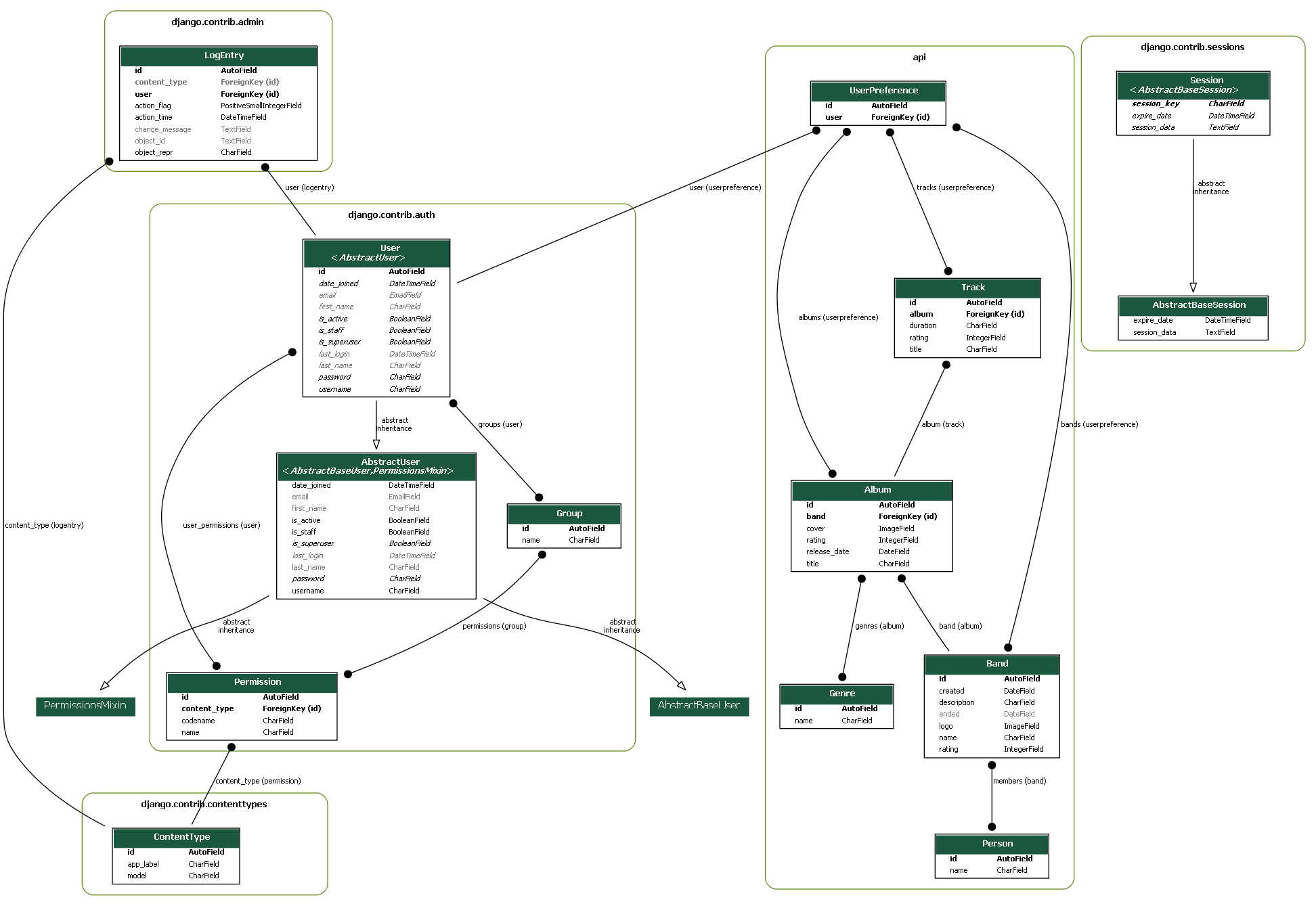
**Задание к лабораторной работе:**

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 выполнения работы:**

Для реализации API разрабатываемого web-приложение используем Django и Django REST Framework.

Структура БД приложения:



**Код разработанного приложения**

Содержимое файла models.py:

from django.db import models

from django.contrib.auth.models import User

from django.conf import settings

class Person(models.Model):

    name = models.CharField(max\_length=255)

class Band(models.Model):

    name = models.CharField(max\_length=255)

    description = models.CharField(max\_length=10000)

    created = models.DateField()

    ended = models.DateField(blank=True, null=True)

    logo = models.ImageField(upload\_to=settings.MEDIA\_ROOT)

    members = models.ManyToManyField(

        'Person'

    )

    rating = models.IntegerField()

    def \_\_str\_\_(self):

        return "Исполнитель: {}".format(self.name)

class Album(models.Model):

    title = models.CharField(max\_length=255)

    release\_date = models.DateField()

    cover = models.ImageField(upload\_to=settings.MEDIA\_ROOT)

    rating = models.IntegerField()

    band = models.ForeignKey(

        'Band',

        on\_delete=models.CASCADE

    )

    genres = models.ManyToManyField(

        'Genre'

    )

    def \_\_str\_\_(self):

        return "Альбом: {}".format(self.title)

class Genre(models.Model):

    name = models.CharField(max\_length=255)

class Track(models.Model):

    title = models.CharField(max\_length=255)

    duration = models.CharField(max\_length=255)

    rating = models.IntegerField()

    album = models.ForeignKey(

        'Album',

        on\_delete=models.CASCADE

    )

    def \_\_str\_\_(self):

        return "{} - {}".format(self.title, self.duration)

class UserPreference(models.Model):

    user = models.ForeignKey(

        User,

        on\_delete=models.CASCADE

    )

    bands = models.ManyToManyField(

        'Band',

        blank=True,

        null=True

    )

    albums = models.ManyToManyField(

        Album,

        blank=True,

        null=True

    )

    tracks = models.ManyToManyField(

        Track,

        blank=True,

        null=True

    )