ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

КУРСОВОЙ ПРОЕКТ

Дисциплина:

«Информационное обеспечение систем управления»

На тему:

«Проектирование реляционной базы данных»

Выполнил: ст. гр. ВУЦ-421

Бобычев В.А.

Проверил: к.т.н., доц. Васильева М.А.

Москва 2024

**ОГЛАВЛЕНИЕ**

[1. Инфологическое проектирование 3](#_Toc168980154)

[1.2 Анализ предметной области 3](#_Toc168980155)

[1.2 Анализ информационных задач и круга пользователей системы 5](#_Toc168980156)

[2. Логическое проектирование реляционной БД 6](#_Toc168980157)

[2.1 Преобразование ER-диаграммы в схему базы данных 6](#_Toc168980158)

[2.2 Составление реляционных отношений 7](#_Toc168980159)

[2.3 Нормализация полученных отношений 10](#_Toc168980160)

[3. Физическое проектирование БД 10](#_Toc168980161)

[3.1 Разработка скриптов на создание базы данных и таблиц 10](#_Toc168980162)

[3.2 Разработка скриптов на добавление данных в таблицы 13](#_Toc168980163)

[3.3 Проверка по готовым запросам 15](#_Toc168980164)

[4. Вывод 19](#_Toc168980165)

# 1. Инфологическое проектирование

# 1.2 Анализ предметной области

База данных создаётся для информационного обслуживания регистрационного отдела поликлиники. БД должна содержать информацию о врачах, ведущих прием, расписании приема, и пациентах, проживающих на определенных участках.

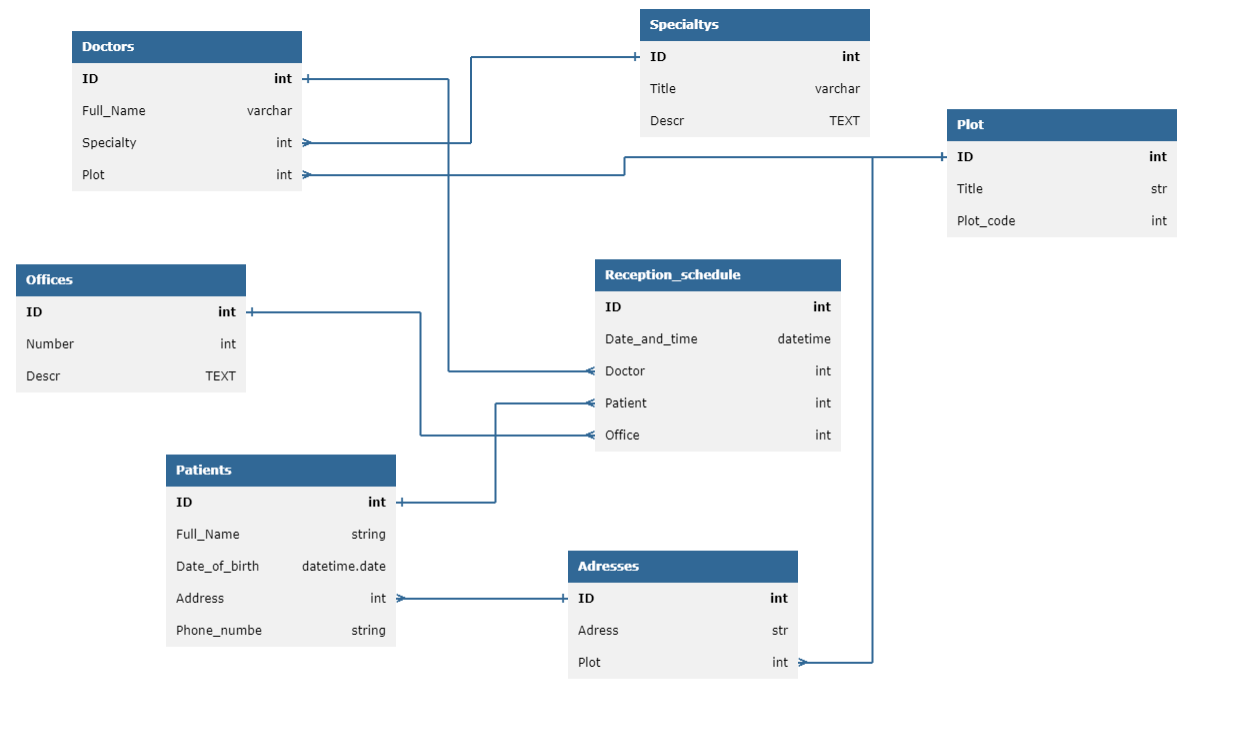
В соответствии с ПрО система строится с учётом следующих особенностей:

1. Доктор может записать пациента
2. Пациент может записаться к доктору
3. У доктора есть специализация
4. Доктор связан с конкретным районом (Адреса в свою очередь связаны с районами)
5. У пациента есть адрес

Выделим базовые сущности этой ПрО:

* Доктор – уникальное ID, ФИО, специализация, район за которым он закреплен;
* Специализация – уникальное ID, название, описание;
* Кабинет – уникальное ID, номер, описание;
* Полки — уникальное ID, адрес, номер полки
* Участок (Район) – уникальное ID, название, индекс участка.
* Пациент – уникальное ID, ФИО, дата рождения, адрес, номер телефона.
* Адреса – уникальное ID, адрес, участок.

Разрабатываемая БД (рисунок 1) является учебной, поэтому сущности были упрощены.



**Рисунок 1 – ER-диаграмма**

# 1.2 Анализ информационных задач и круга пользователей системы

Система создается для обслуживания домашней библиотеки.

Определим границы информационной поддержки пользователей:

1. *Функциональные возможности:*

* ведение БД (запись, чтение, модификация, удаление в архив);
* обеспечение логической непротиворечивости БД;
* обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
* реализация наиболее часто встречающихся запросов в готовом виде.

1. *Готовые запросы:*

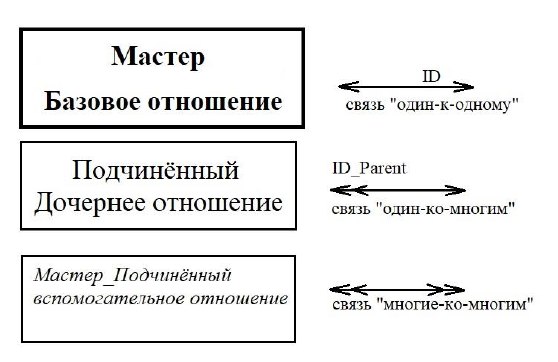
* Получение списка всех врачей с их специальностями.
* Получение списка пациентов, родившихся после определенной даты.
* Подсчет количества приемов у каждого врача.
* Получение списка пациентов и их адресов.
* Получение расписания приемов в определенном офисе по его номеру.
* Получение списка врачей, работающих на определенном участке.
* Получение списка пациентов с их номерами телефонов, отсортированных по дате рождения.
* Получение списка офисов с описанием, не имеющих запланированных приемов.
* Получение списка пациентов, у которых есть приемы у врачей определенной специальности.
* Получение списка всех приемов, запланированных на следующую неделю.

# 2. Логическое проектирование реляционной БД

# 2.1 Преобразование ER-диаграммы в схему базы данных

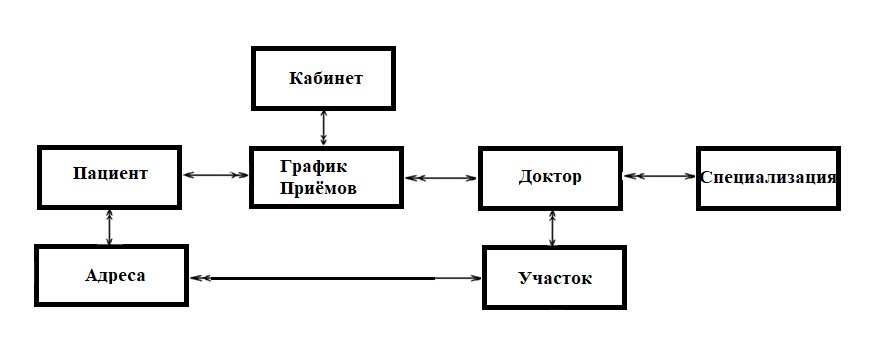
База данных создаётся на основании схемы базы данных. Преобразование ER-диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД).

На схеме, представленной ниже (Рисунок 2), введены обозначения:



**Рисунок 2 – Обозначения, используемые на схеме базы данных**

Полученная схема реляционной базы данных (РБД) приведена ниже (Рисунок 3).



**Рисунок 3 – Схема РБД, полученная из ER-диаграммы**

Связь типа 1:n («один-ко-многим») между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь.

# 2.2 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту предметной области) и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). В том случае, если базовое отношение не имеет потенциальных ключей, вводится суррогатный первичный ключ, который не несёт смысловой нагрузки и служит только для идентификации записей.

Введем для всех сущностей ключевой атрибут - суррогатный ключ ID.

Отношения приведены ниже (см. Таблица 1 – Таблица 5). Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной.

**Таблица 1 – Схема отношения Доктора (Doctors)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| ФИО | Full\_Name | Строка (255) |  |
| Специализация | Specialty | Целое число | Внешний ключ (Specialty.ID) |
| Участок | Plot | Целое число | Внешний ключ (Plot.ID) |

**Таблица 2 – Схема отношения Пациенты (Patients)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| ФИО | Full\_Name | Строка (255) |  |
| Дата рождения | Date\_of\_birth | Дата |  |
| Адрес | Address | Целое число | Внешний ключ (Addresses.ID) |
| Номер телефона | Phone\_number | Строка (20) |  |

**Таблица 3 – Схема отношения Кабинеты (Offices)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| Номер кабинета | Number | Целое число |  |
| Описание | Descr | Текст |  |

**Таблица 4 – Схема отношения Специальности (**Specialtys**)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| Название | Title | Строка (255) |  |
| Описание | Descr | Текст |  |

**Таблица 5 – Схема отношения Очередь записи (Reception\_schedule)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| Дата и время записи | Date\_and\_time | datetime |  |
| Доктор | Doctor | Целое число | Внешний ключ (Doctors.ID) |
| Пациент | Patient | Целое число | Внешний ключ (Patients.ID) |
| Кабинет | Office | Целое число | Внешний ключ (Offices.ID) |

**Таблица 6 – Схема отношения Адреса Пациентов (Adresses)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| Адрес | Adress | Строка (255) |  |
| Участок | Plot | Целое число | Внешний ключ (Plot.ID) |

**Таблица 7 – Схема отношения Участки (Plot)**

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | Первичный ключ |
| Название | Title | Строка (255) |  |
| Индекс участка | Plot\_code | Целое число |  |

# 2.3 Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.

В нашей спроектированной БД 1НФ выполняется без создания новых таблиц, в которых были разбиты сложные атрибуты на простые, так как заранее всё было выполнено.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

**3НФ.** Все отношения находятся во 3НФ, так как в каждом из них отсутсвуют транзитивные функциональные зависимости.

# 3. Физическое проектирование БД

# 3.1 Разработка скриптов на создание базы данных и таблиц

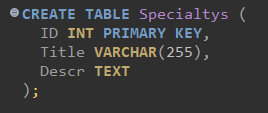
*Создание базы данных «**Polyclinic»:*

**create** **database** **"** **Polyclinic "**;

*Создание таблицы «Specialtys»:*

Добавление первичного ключа

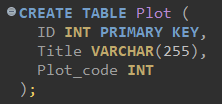
Добавление ограничения количества символов поля Title



*Создание таблицы «Plot»:*

Добавление первичного ключа:

Добавление ограничения количества символов поля Title



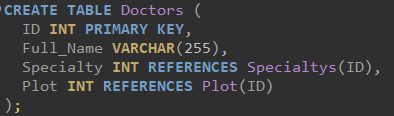
*Создание таблицы «Doctors»:*

Добавление первичного ключа:

Добавление ограничения количества символов поля Full\_Name

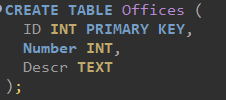
Добавление внешнего ключа на *Specialtys.ID*

Добавление внешнего ключа на *Plot.ID*



*Создание таблицы «Offices»:*

Добавление первичного ключа:

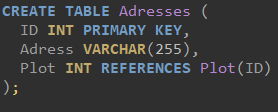


*Создание таблицы «Adresses»:*

Добавление первичного ключа:

Добавление ограничения количества символов поля Adress

Добавление внешнего ключа на *Plot.ID*



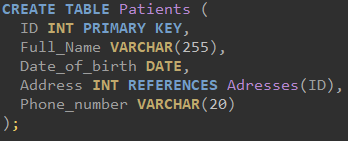
*Создание таблицы «Patients»:*

Добавление первичного ключа:

Добавление ограничения количества символов поля Full\_Name

Добавление внешнего ключа на *Adresses.ID*

Добавление ограничения количества символов поля Phone\_number



*Создание таблицы «Patients»:*

Добавление первичного ключа:

Добавление внешнего ключа на *Doctors.ID*

Добавление внешнего ключа на *Patients.ID*

Добавление внешнего ключа на *Offices.ID*

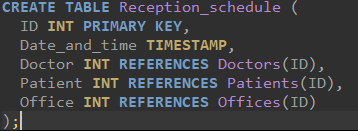
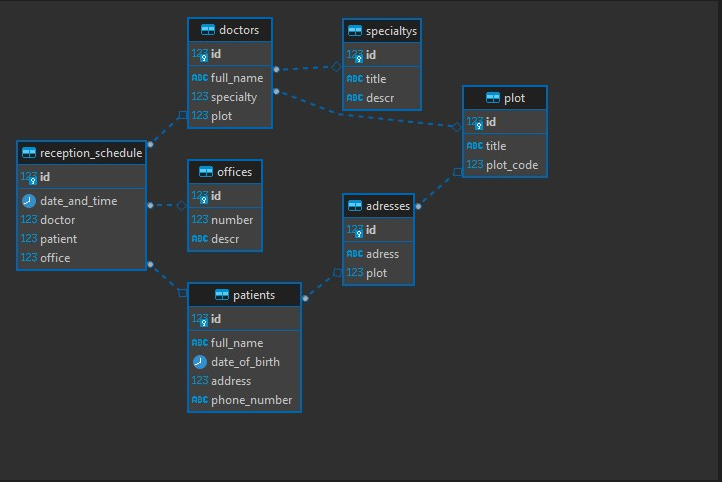


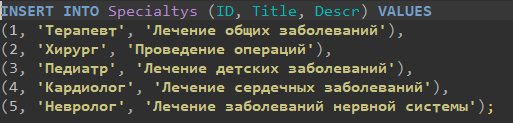
Диаграмма базы данных представлена ниже (Рисунок 4).



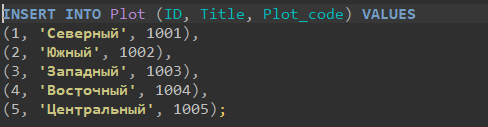
**Рисунок 4 – Диаграмма базы данных**

# 3.2 Разработка скриптов на добавление данных в таблицы

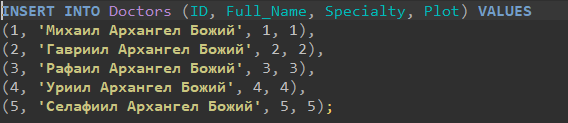
*Заполнение таблицы Specialtys:*



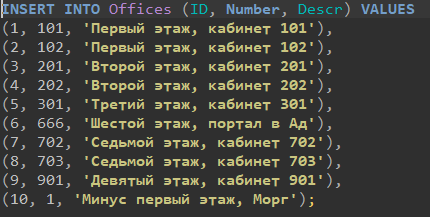
*Заполнение таблицы Plot:*



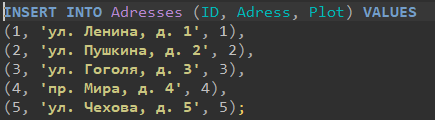
*Заполнение таблицы Doctors*



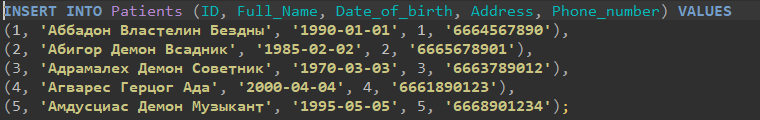
*Заполнение таблицы Offices*



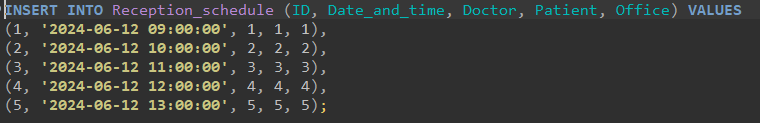
*Заполнение таблицы Adresses*



*Заполнение таблицы Patients:*

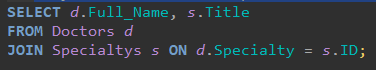


*Заполнение таблицы Reception\_schedule:*

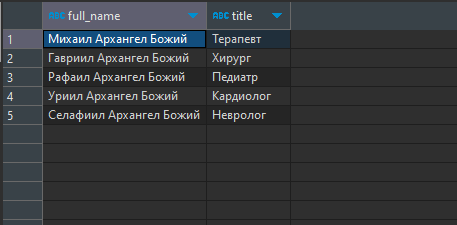


# 3.3 Проверка по готовым запросам

*Получение списка всех врачей с их специальностями:*

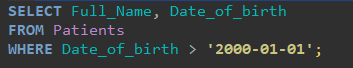


Результат представлен на Рисунке 5.

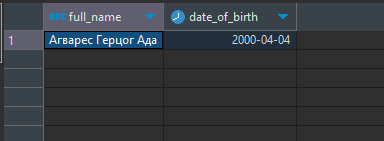


**Рисунок 5 – Результат работы скрипта**

*Получение списка пациентов, родившихся после определенной даты:*

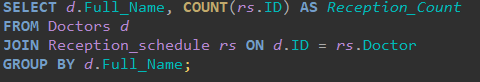


Результат представлен на Рисунке 6.

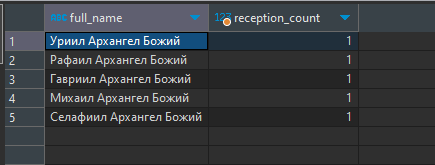


**Рисунок 6 – Результат работы скрипта**

*Подсчет количества приемов у каждого врача:*

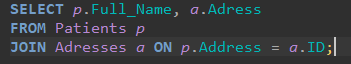


Результат представлен на Рисунке 7.

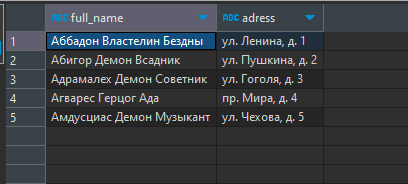


**Рисунок 7 – Результат работы скрипта**

*Получение списка пациентов и их адресов:*

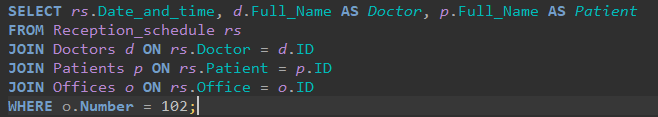


Результат представлен на Рисунке 8.

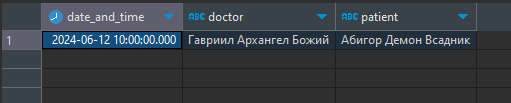


**Рисунок 8 – Результат работы скрипта**

*Получение расписания приемов в определенном офисе по его номеру:*

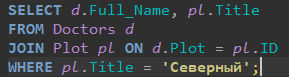


Результат представлен на Рисунке 9.

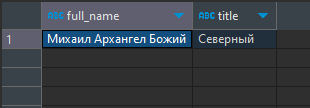


**Рисунок 9 – Результат работы скрипта**

*Получение списка врачей, работающих на определенном участке:*



Результат представлен на Рисунке 10.

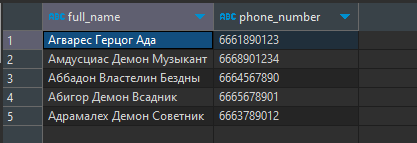


**Рисунок 10 – Результат работы скрипта**

*Получение списка пациентов с их номерами телефонов, отсортированных по дате рождения:*

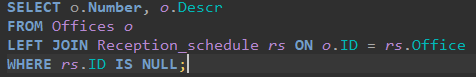


Результат представлен на Рисунке 11.

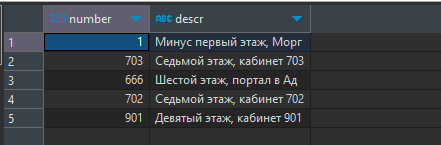


**Рисунок 11 – Результат работы скрипта**

*Получение списка офисов с описанием, не имеющих запланированных приемов:*

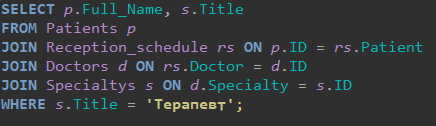


Результат представлен на Рисунке 12.

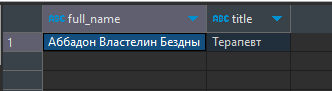


**Рисунок 12 – Результат работы скрипта**

*Получение списка пациентов, у которых есть приемы у врачей определенной специальности:*

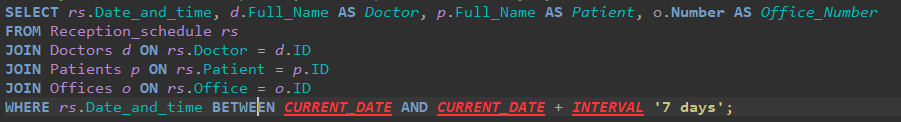


Результат представлен на Рисунке 13.

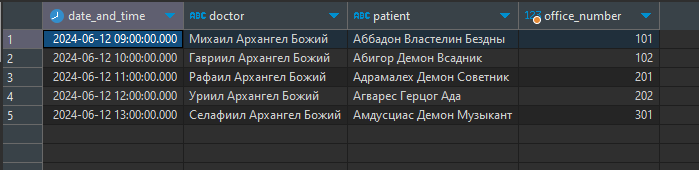


**Рисунок 13 – Результат работы скрипта**

*Получение списка всех приемов, запланированных на следующую неделю:*



Результат представлен на Рисунке 14.



**Рисунок 14 – Результат работы скрипта**

# 4. Вывод

Были изучены методы и закреплены знания в проектировании реляционных баз данных (РБД) в системе управления базами данных (СУБД) PostgresSQL.