МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Информатика и системы управления»

«Проектирование базы данных MySQL»

ОТЧЕТ

по лабораторной работе № 1

по дисциплине

\_\_\_\_\_\_\_\_\_\_ Управление данными\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Балашова Т.И. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Опарин С. А.

(подпись) (фамилия, и.,о.)

17-СБК

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород   
2020

**Задание:**

1. Построить ER – диаграмму и дать её краткое описание. По полученной ER – диаграмме построить схему данных. Создать таблицы БД при помощи PostgreSQL. Определить отношения между полученными таблицами.

2.Выполнить базовые операции работы с таблицами:

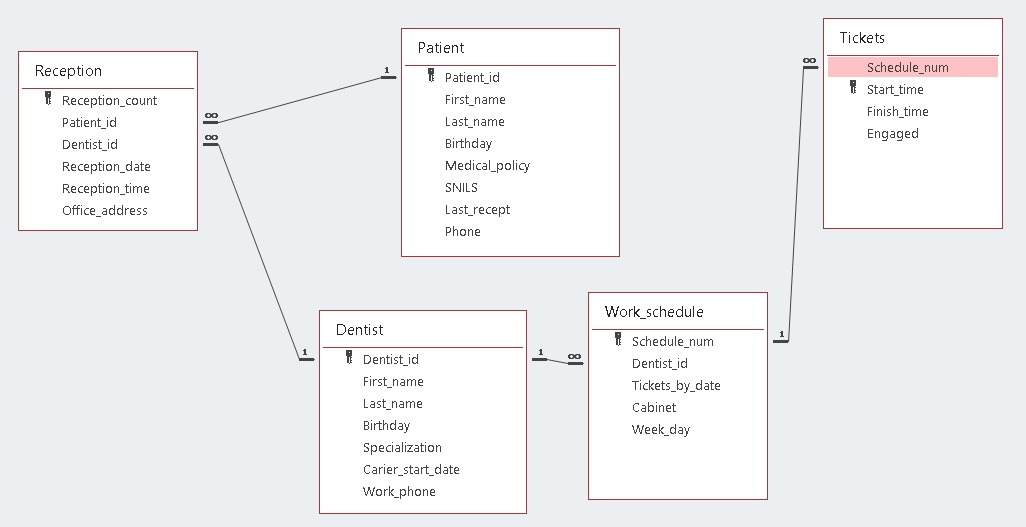
просмотр таблиц, доступных в БД, создать новую таблицу в БД, добавление новых данных в таблицу, переименовать таблицу, отсортировать таблицу, удалить все данные из таблицы, удалить таблицу из БД.

3. На основе данных предыдущих лабораторных работ создать следующие запросы: условный запрос, вложенный запрос, запрос с использованием объединения таблиц, запрос с группировкой, запрос с использованием агрегирующей функцией.

Отчет должен содержать: ER модель базы данных, скрипты создания, наполнение таблиц тестовыми данными, SQL-запросы, базовые операции работы с таблицами и скриншоты их выполнения

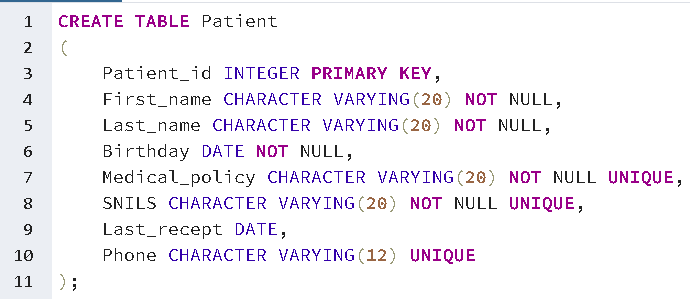
**Схема БД:**

**Предметна область: «Cтоматологическая клиника»**

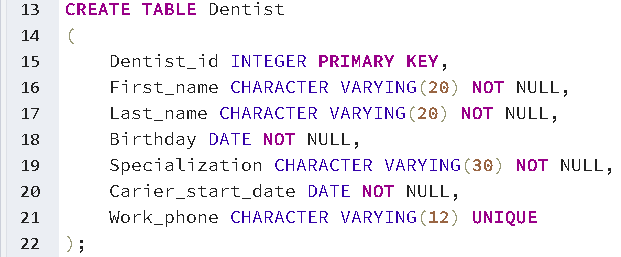


* **Создание таблицы:**

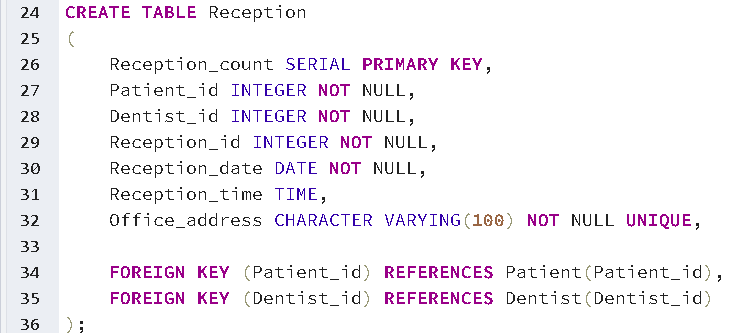
1. **Пациент (Patient)**

****

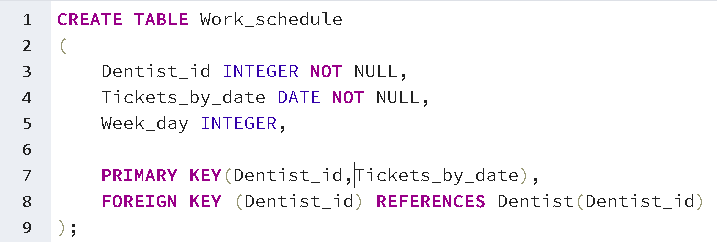
1. **Стоматолог (Dentist)**

****

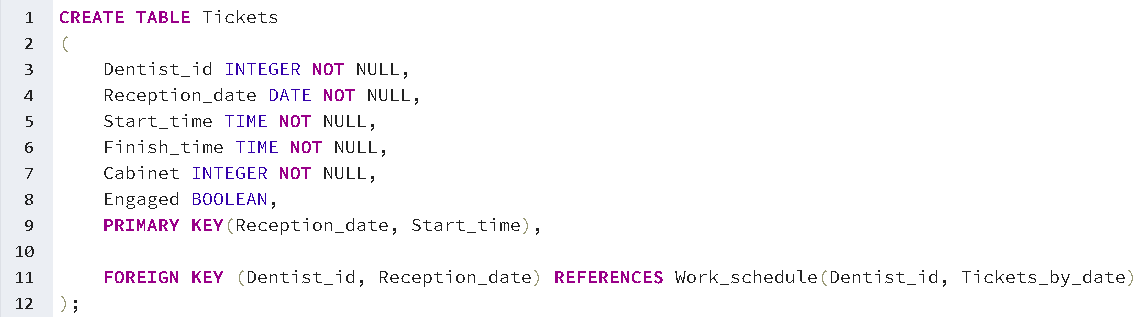
1. **Приём (Reception)**

****

1. **Расписание (Work\_schedule)**

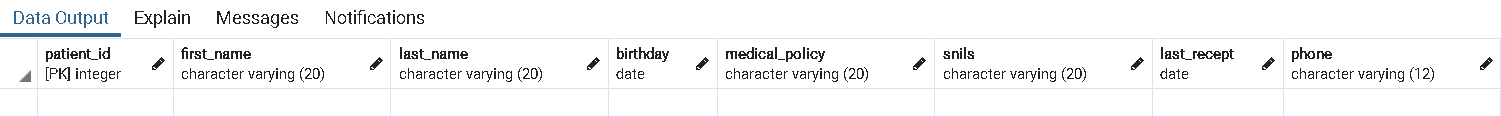
****

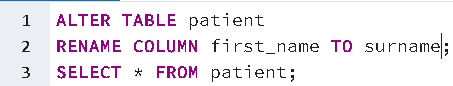
1. **Талоны (Tickets)**



* **Переименование столбца таблицы:**

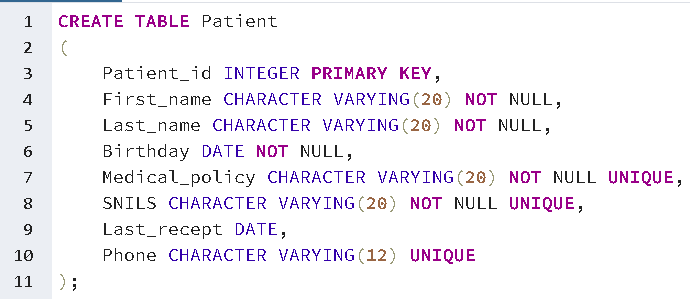
****



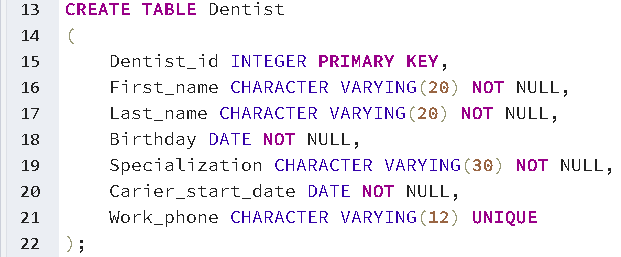




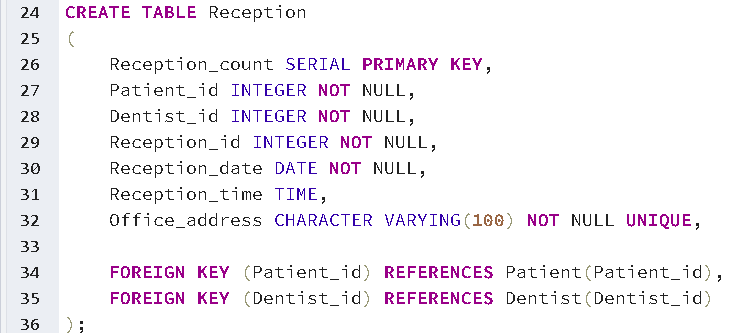
* **Добавление новых данных в таблицу:**
* **Пациент (Patient)**

****

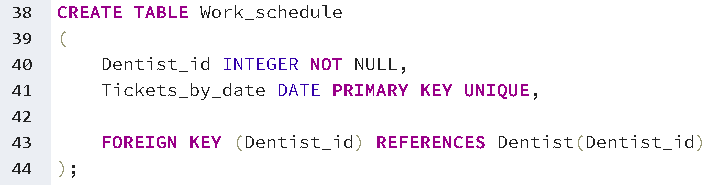
* **Стоматолог (Dentist)**

****

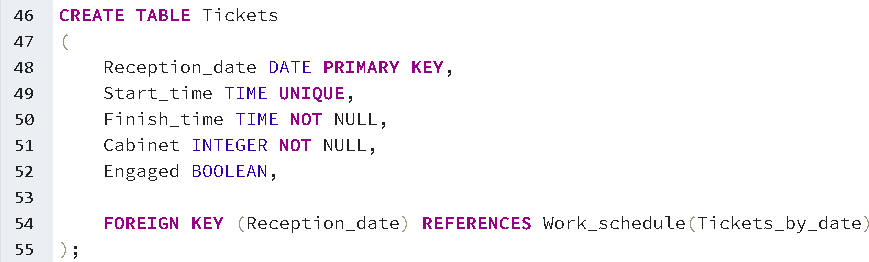
* **Приём (Reception)**

****

* **Расписание (Work\_schedule)**

****

* **Талоны (Tickets)**



1. **Пациент (Patient)**

INSERT INTO Patient(patient\_id, surname, last\_name, birthday, medical\_policy, snils, last\_recept, phone)

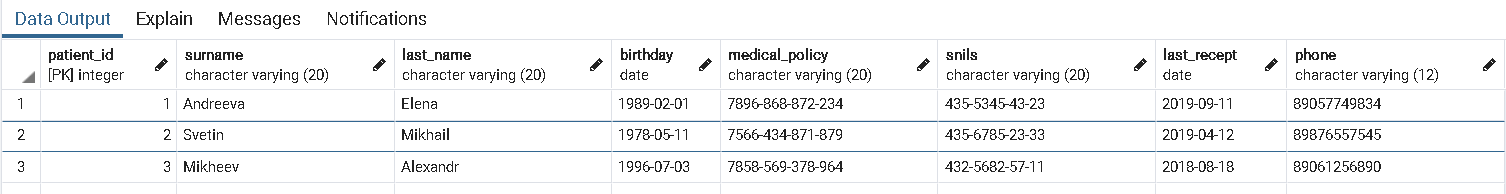
VALUES

(1, 'Andreeva', 'Elena', '1989-02-01', '7896-868-872-234', '435-5345-43-23', ' 2019-09-11', '89057749834'),

(2, 'Svetin', 'Mikhail', '1978-05-11', '7566-434-871-879', '435-6785-23-33', '2019-04-12', '89876557545'),

(3, 'Mikheev', 'Alexandr', '1996-07-03', '7858-569-378-964', '432-5682-57-11', '2018-08-18', '89061256890');

SELECT \* FROM Patient;



1. **Стоматолог (Dentist)**

INSERT INTO Dentist(dentist\_id, first\_name, last\_name, birthday, specialization, carier\_start\_date, work\_phone)

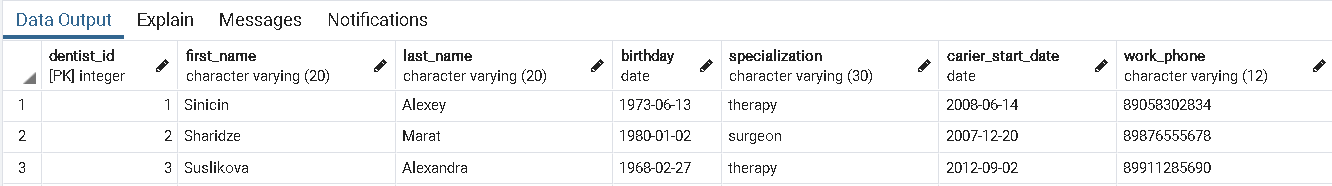
VALUES

(1, 'Sinicin', 'Alexey', '1973-06-13', 'therapy', '2008-06-14', '89058302834'),

(2, 'Sharidze', 'Marat', '1980-01-02', 'surgeon', '2007-12-20', '89876555678'),

(3, 'Suslikova', 'Alexandra', '1968-02-27', 'therapy', '2012-09-02', '89911285690');

SELECT \* FROM Dentist;



1. **Приём (Reception)**

INSERT INTO Reception(reception\_id, patient\_id, dentist\_id, reception\_date, reception\_time, office\_address)

VALUES

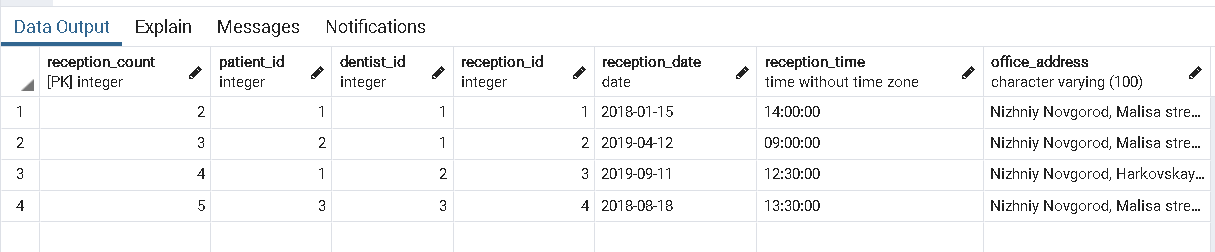
(1, 1, 1, '2018-01-15', '14:00:00', 'Nizhniy Novgorod, Malisa street, 25'),

(2, 2, 1, '2019-04-12', '09:00:00', 'Nizhniy Novgorod, Malisa street, 25'),

(3, 1, 2, '2019-09-11', '12:30:00', 'Nizhniy Novgorod, Harkovskaya street, 4'),

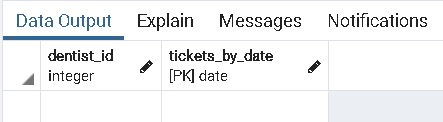
(4, 3, 3, '2018-08-18', '13:30:00', 'Nizhniy Novgorod, Malisa street, 25');

SELECT \* FROM Reception;



1. **Расписание (Work\_schedule)**

****

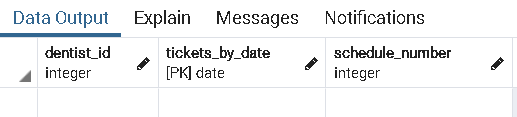
****

**Добавим столбец «schedule\_number» и внесём данные**

ALTER TABLE work\_schedule

ADD schedule\_number INTEGER NULL;

SELECT \* FROM work\_schedule;

.

INSERT INTO work\_schedule(dentist\_id, Tickets\_by\_date)

VALUES

(1, '2020-03-12'),

(2, '2020-03-12'),

(3, '2020-03-12'),

(1, '2020-03-13'),

(2, '2020-03-13'),

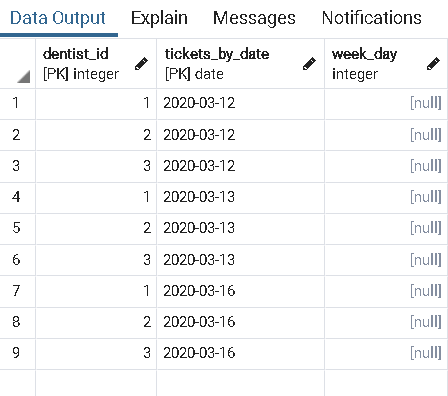
(3, '2020-03-13'),

(1, '2020-03-16'),

(2, '2020-03-16'),

(3, '2020-03-16');

SELECT \* FROM work\_schedule;



1. **Талоны (Tickets)**

INSERT INTO tickets(dentitst\_id, reception\_date, start\_time, finish\_time, cabinet, engaged)

VALUES

(1, '2020-03-16', '09:00:00', '09:25:00', 5, 0),

(1, '2020-03-16', '09:30:00', '09:55:00', 5, 0),

(1, '2020-03-16', '10:00:00', '10:25:00', 5, 1),

(1, '2020-03-16', '10:30:00', '10:55:00', 5, 1),

(1, '2020-03-16', '11:00:00', '11:25:00', 5, 1),

(1, '2020-03-16', '11:30:00', '11:55:00', 5, 1),

(1, '2020-03-16', '12:00:00', '12:25:00', 5, 0),

(1, '2020-03-16', '12:30:00', '12:55:00', 5, 0),

(3, '2020-03-16', '14:00:00', '04:25:00', 6, 0),

(3, '2020-03-16', '14:30:00', '04:55:00', 6, 1),

(3, '2020-03-16', '15:00:00', '15:25:00', 6, 1),

(3, '2020-03-16', '15:30:00', '15:55:00', 6, 0),

(3, '2020-03-16', '16:00:00', '16:25:00', 6, 0),

(3, '2020-03-16', '16:30:00', '16:55:00', 6, 0),

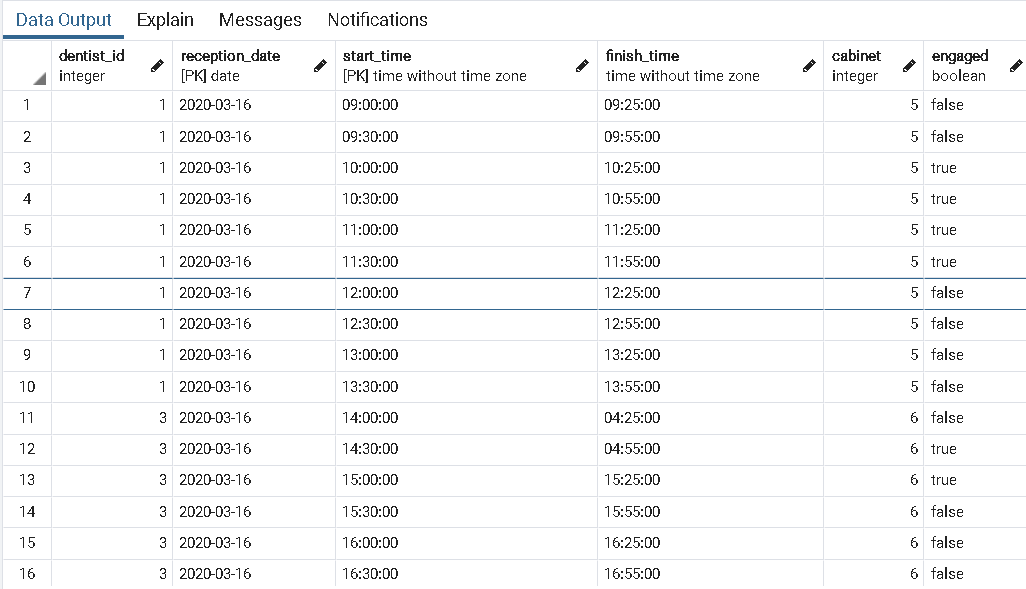
(3, '2020-03-16', '17:00:00', '17:25:00', 6, 1),

(3, '2020-03-16', '17:30:00', '17:55:00', 6, 1),

(3, '2020-03-16', '18:00:00', '18:25:00', 6, 1),

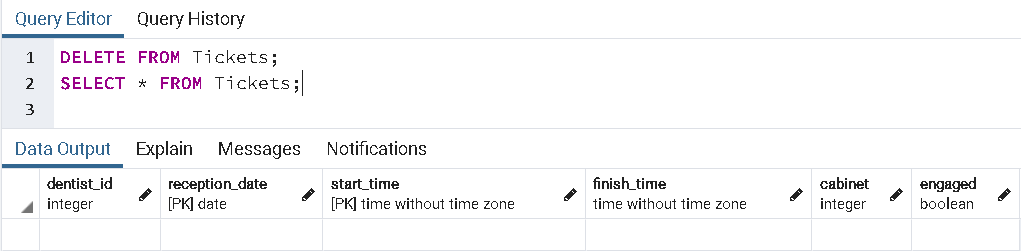
(3, '2020-03-16', '18:30:00', '18:55:00', 6, 0);

SELECT \* FROM tickets;



* **Удаление всех данных из таблицы:**

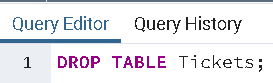
**Будут удалены, а потом восстановлены данные из предыдущего пункта.**



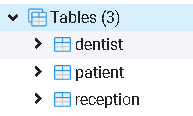
Замечание:

Таблиц, не являющуюся листовой, удалить командой DELETE не получится, пока не будут удалены все таблицы, участвующие в связях.

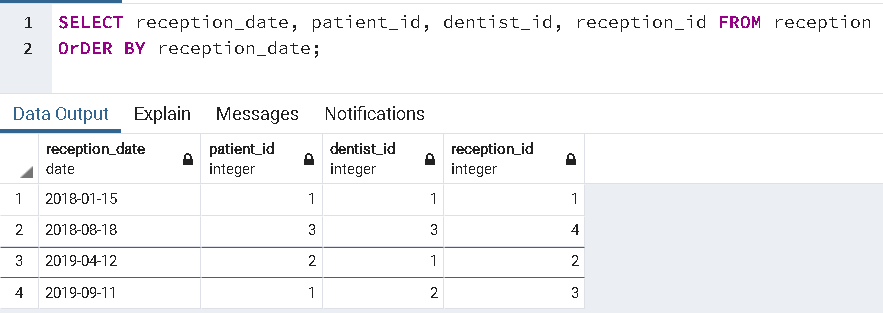
* **Удаление таблицы из БД:**

**** // мы вынуждены удалить«лист», чтобы удалить «ветку»

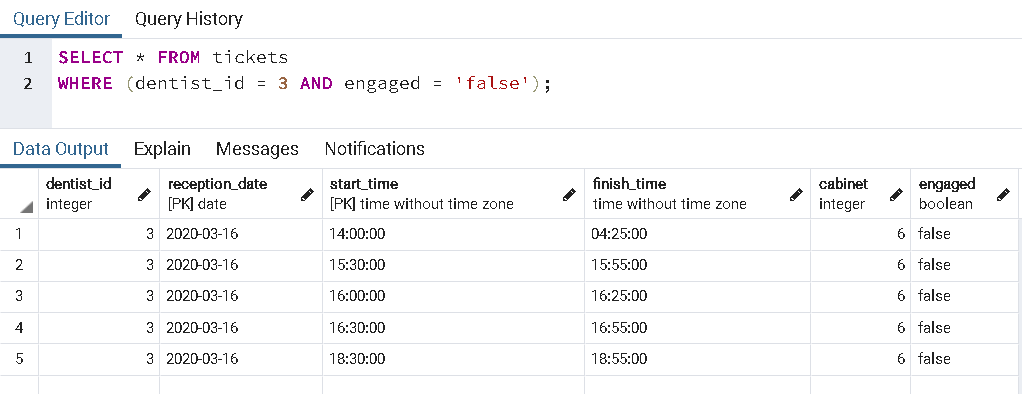




* **Cортировка таблицы:**

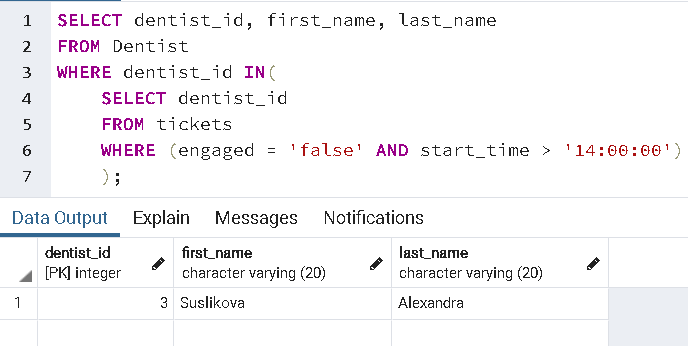
****

* **Условный запрос c применением булевого оператора**



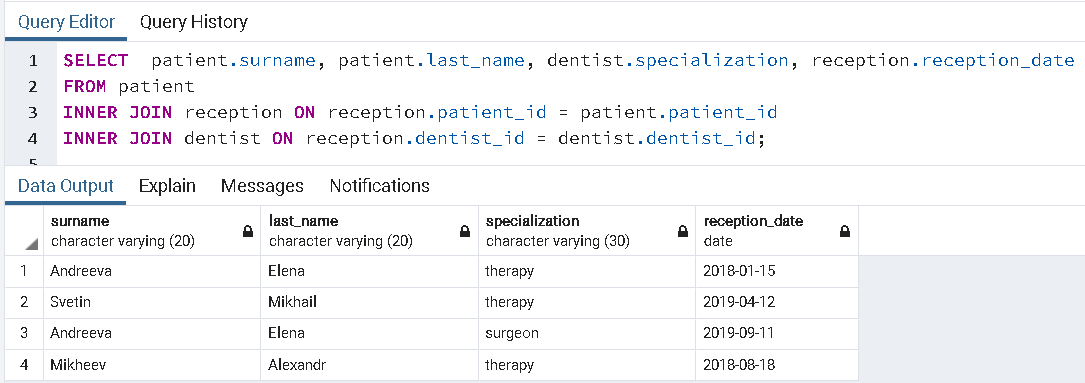
Данный запрос отображает свободные талоны конкретного стоматолога.

* **Вложенный запрос:**



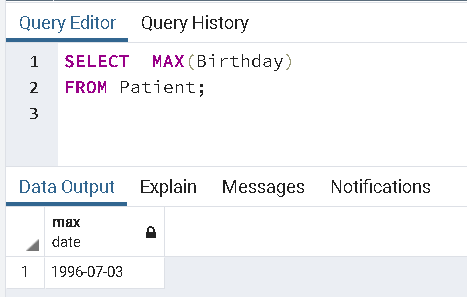
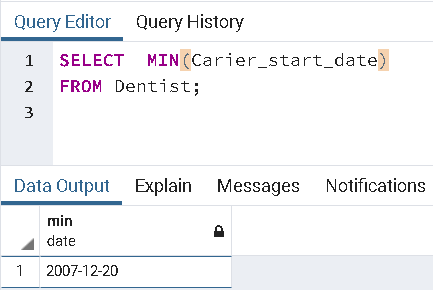
Данный запрос отображает специалистов со свободными местами до 14.00.

* **Запрос с использованием объединения таблиц:**



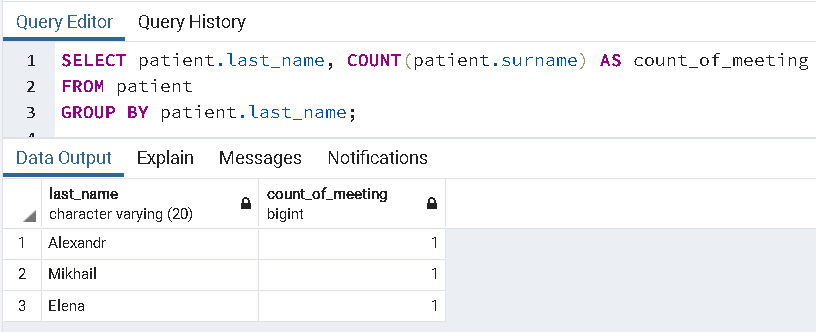
Объединено 3 таблицы для вывода проведённых услуг по датам.

* **Запрос с использованием агрегирующей функцией:**

Самый молодой посетитель. Самый опытный сотрудник.

* **Запросы с группировкой:**



C использованием HAVING:

