**ОТЧЕТ**

**О ЛАБОРАТОРНОЙ РАБОТЕ № 5**

По теме: Создание SQL-запросов в Pgadmin

По дисциплине: Основы проектирования баз данных

Специальность 09.02.07 «Информационные системы и программирование»

УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Санкт-Петербург 2020

Преподаватель:

\_\_\_\_\_\_\_\_Говоров А.И.

Дата: «\_\_» \_\_\_\_\_\_\_ 20\_\_г.

Оценка \_\_\_\_\_\_\_

Выполнил:

Студент группы № Y2339

\_\_\_\_\_\_\_\_\_Карелова А.С.

**Цель работы:** овладеть практическими навыками создания некоторого количества SQL-запросов.

**Порядок выполнения работы:** познакомиться с инструкциями по работе с отладчиком, написать программу по полученному заданию и, выполняя ее по командам, заполнить таблицы в отчете. Операции умножения и деления выполнять через сложение и вычитание в виде отдельных процедур. Вызов и возврат из процедуры выполнять через команды переходов. При делении учитывать остаток.

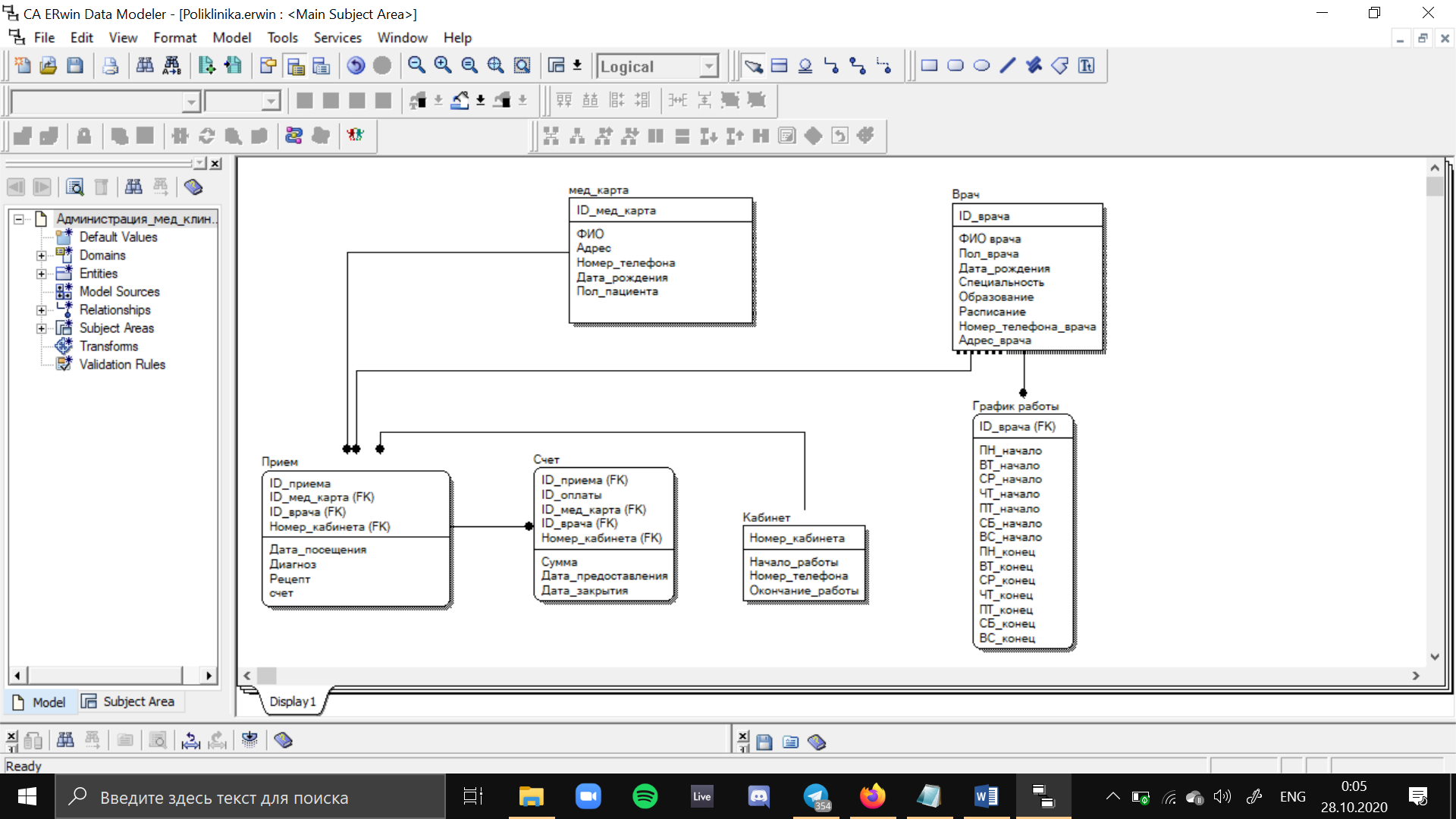
**Программное обеспечение:** пакеты лицензионного ПО общего назначения.

**Практическое задание:**

1. выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой;
2. использование условий WHERE, состоящих из более, чем одного условия;
3. использование функций для работы с датами;
4. использование строковых функций;
5. запрос с использованием подзапросов;
6. вычисление групповой (агрегатной) функции;
7. вычисление групповой (агрегатной) функции с условием HAVING;
8. использование предикатов EXISTS, ALL, SOME и ANY;
9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.);
10. использование объединений запросов (inner join и т.д.).

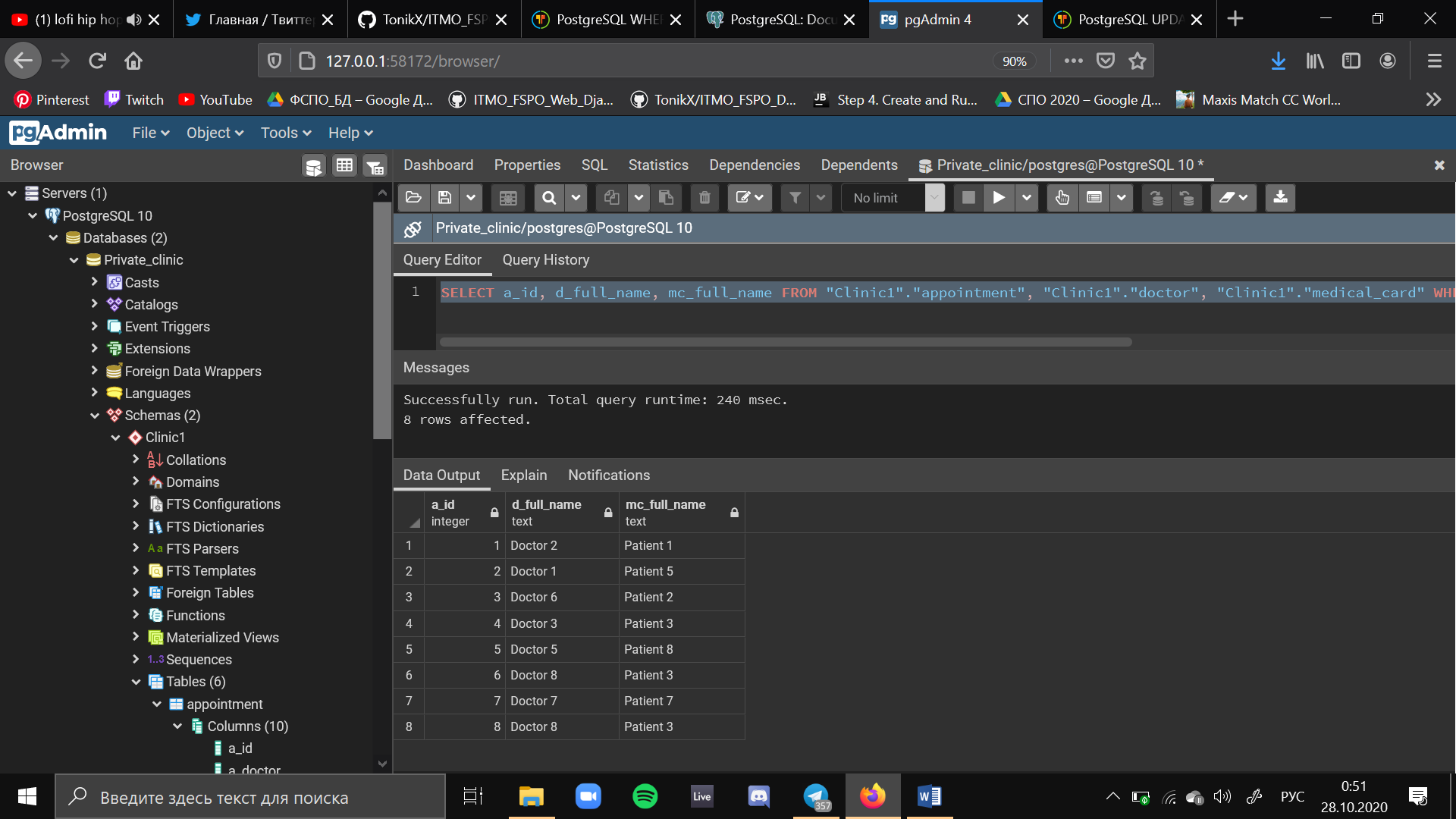
**Выполнение задания:**

На рисунке 1 представлена разработанная ранее система, предназначенная для работников частной клиники.

  
*Рисунок 1 - Модель БД*

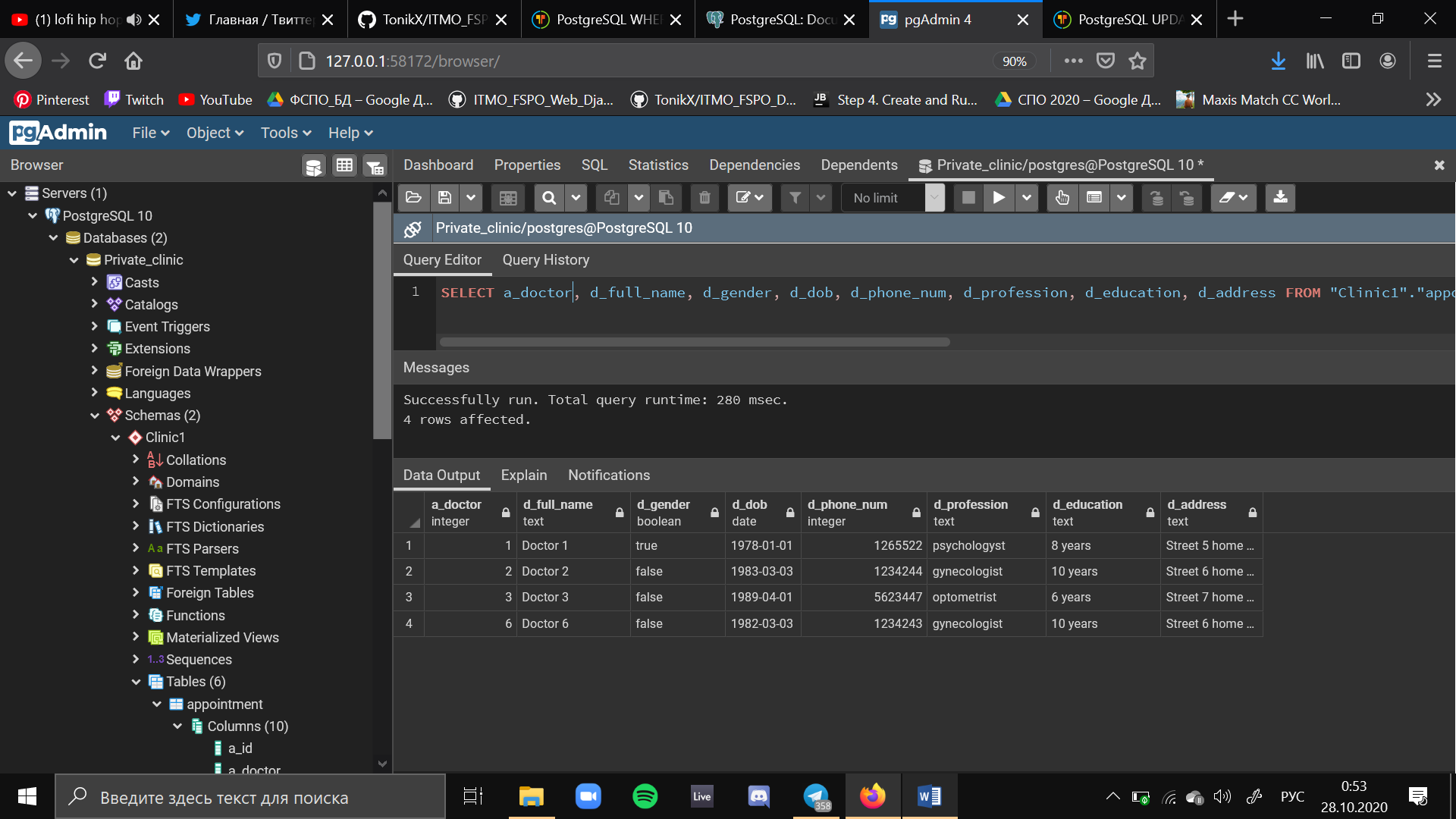
1. На рисунке 2 представлены полные имена врачей и пациентов, которые были в определенные приемы, организованные по возростанию ID приема.

SELECT a\_id, d\_full\_name, mc\_full\_name FROM "clinic"."appointment", "clinic"."doctor", "clinic"."medical\_card" WHERE a\_doctor = d\_id AND a\_patient = mc\_id ORDER BY a\_id;

  
*Рисунок 2 - Результат запроса*

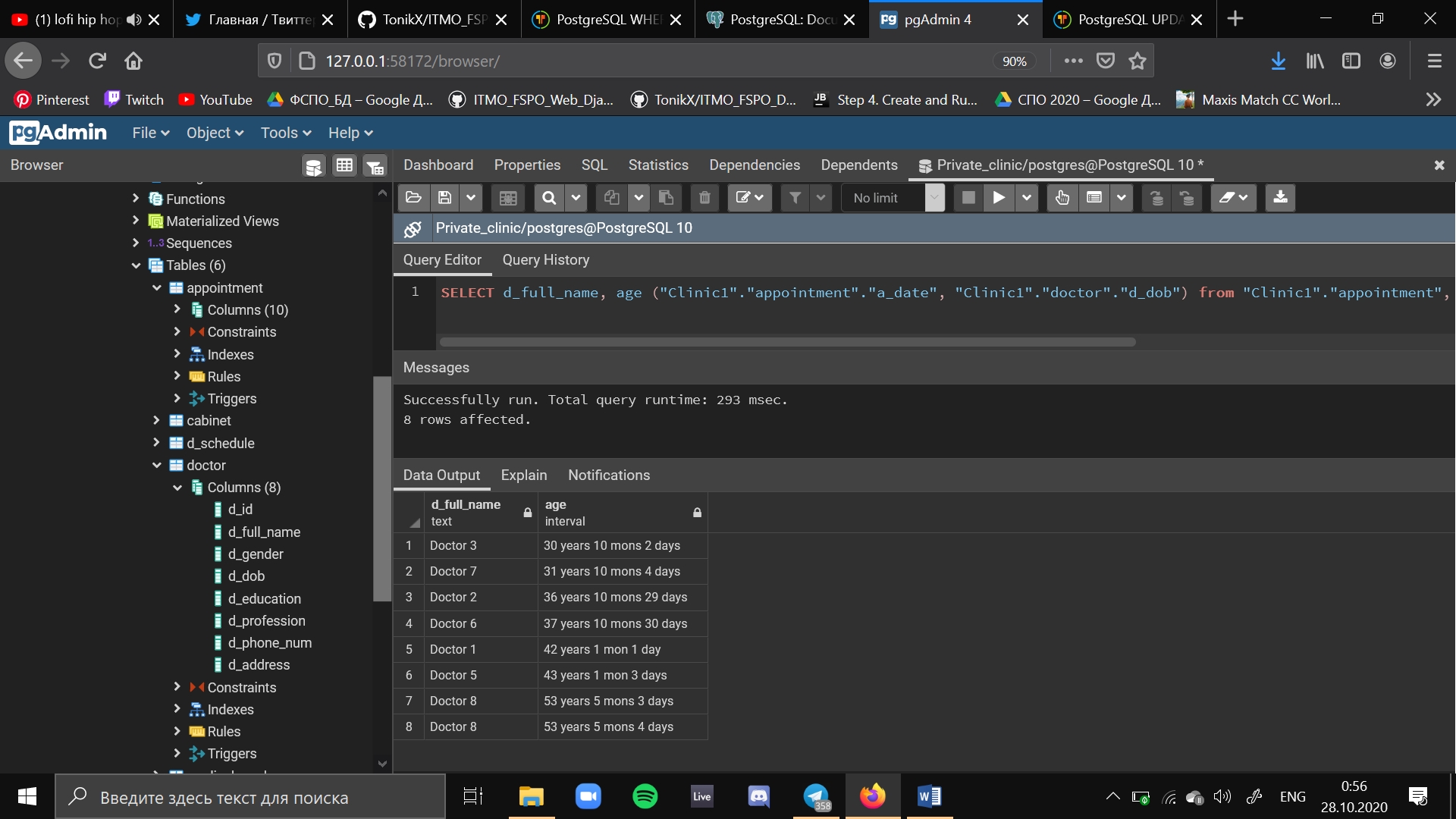
1. На рисунке 3 представлена информация о врачах, которые принимали пациентов в период с 01.02.2020 по 03.02.2020, организованная по ID врача.

SELECT a\_doctor, d\_full\_name, d\_gender, d\_dob, d\_phone\_num, d\_profession, d\_education, d\_address FROM "clinic"."appointment", "clinic"."doctor" WHERE a\_doctor = d\_id AND a\_date BETWEEN '2020-02-01' and '2020-02-03' ORDER BY a\_doctor;

  
*Рисунок 3 - Результат запроса*

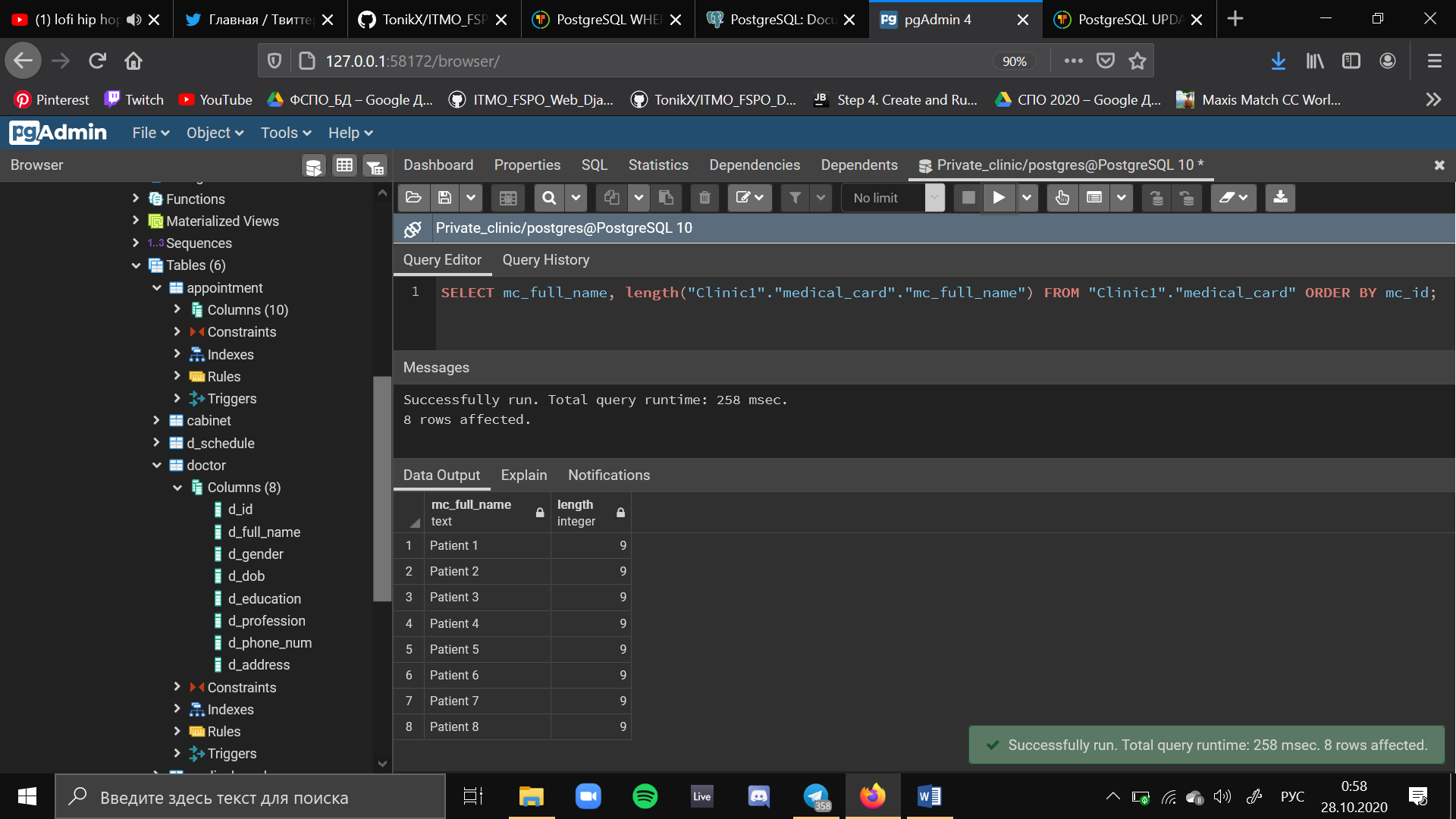
1. На рисунке 4 представлена информация об интервале между датой приема и датой рождения врача, организованная по возрастанию разницы (показывает возраст врача на дату приема).

SELECT d\_full\_name, age ("clinic"."appointment"."a\_date", "clinic"."doctor"."d\_dob") from "clinic"."appointment", "clinic"."doctor" WHERE a\_doctor = d\_id ORDER BY age;

*Рисунок 4 - Результат запроса*

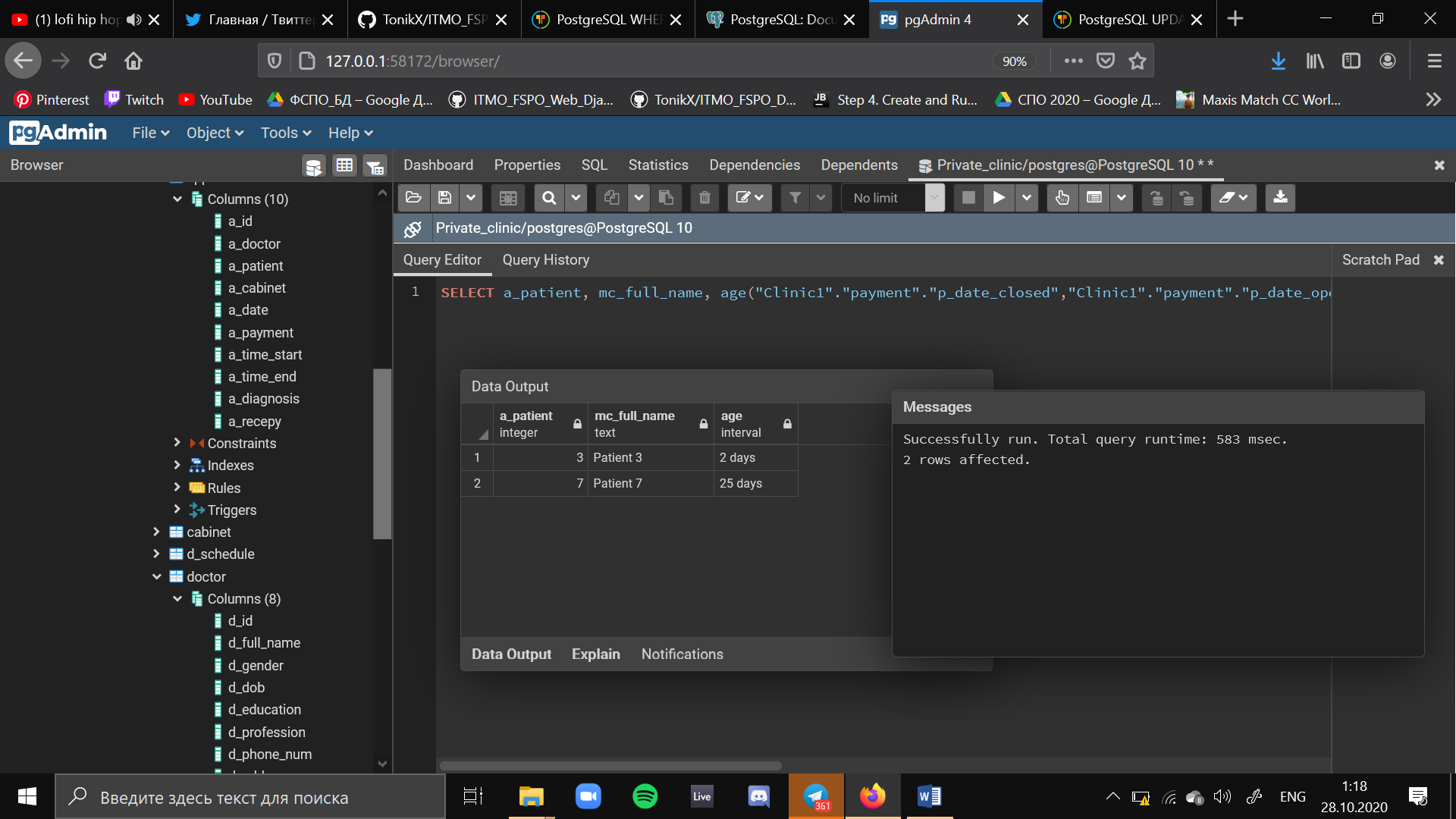
1. На рисунке 5 представлена информация о количестве символов в полных именах пациентов, организованно по возрастанию номера ID пациента.

SELECT mc\_full\_name, length("clinic"."medical\_card"."mc\_full\_name") FROM "clinic"."medical\_card" ORDER BY mc\_id;

*Рисунок 5 - Результат запроса*

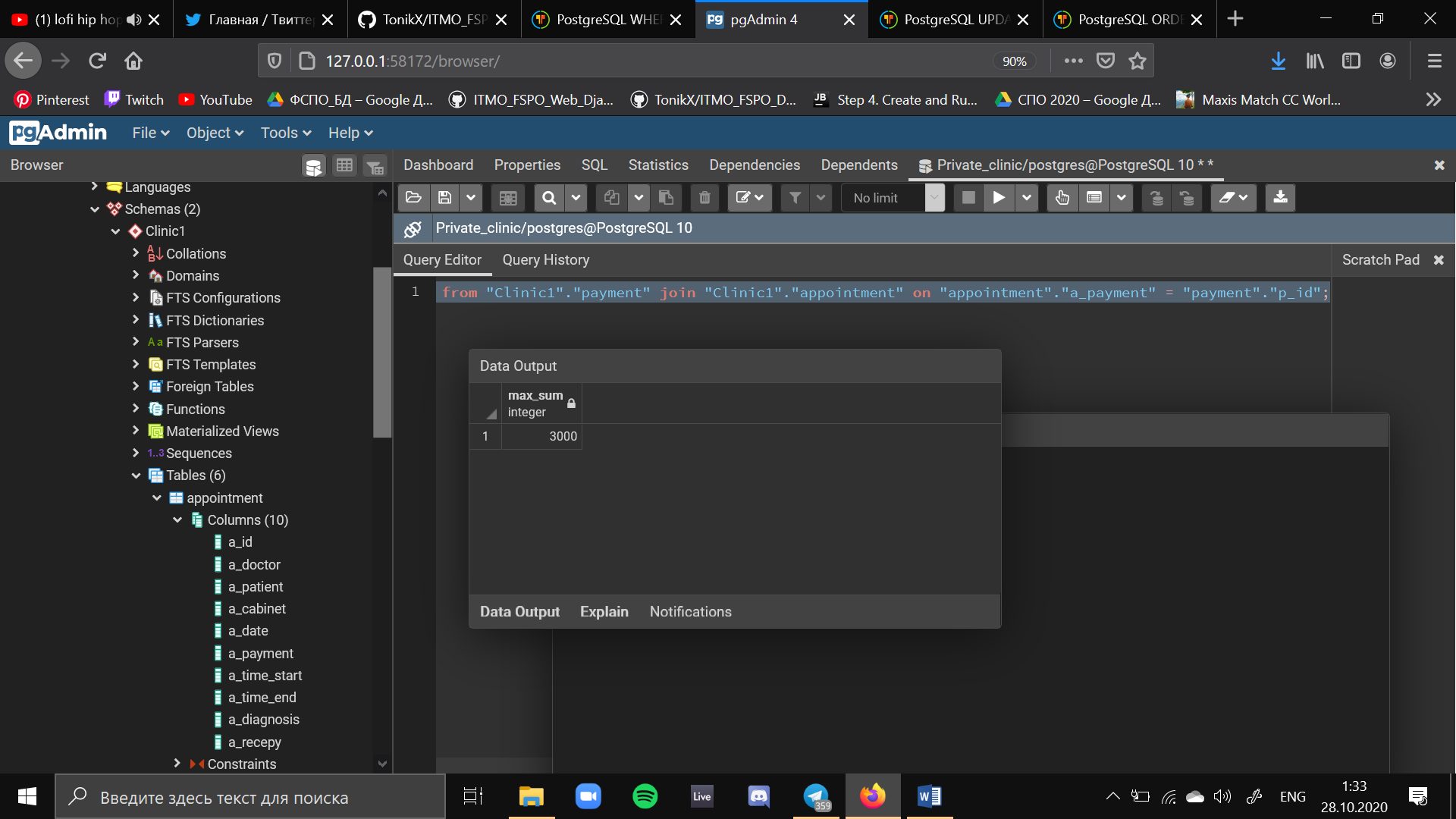
1. На рисунке 6 представлена информация пациентов с диагнозом miosis: id, имена пациентов и то, как долго они не оплачивали; организованно по возрастанию id пациента.

SELECT a\_patient, mc\_full\_name, age("clinic"."payment"."p\_date\_closed","clinic"."payment"."p\_date\_opened") FROM "clinic"."payment", "clinic"."medical\_card", "clinic"."appointment" WHERE a\_diagnosis = 'miosis' AND a\_patient = mc\_id AND a\_payment = p\_id ORDER BY mc\_id;

  
*Рисунок 6 - Результат запроса*

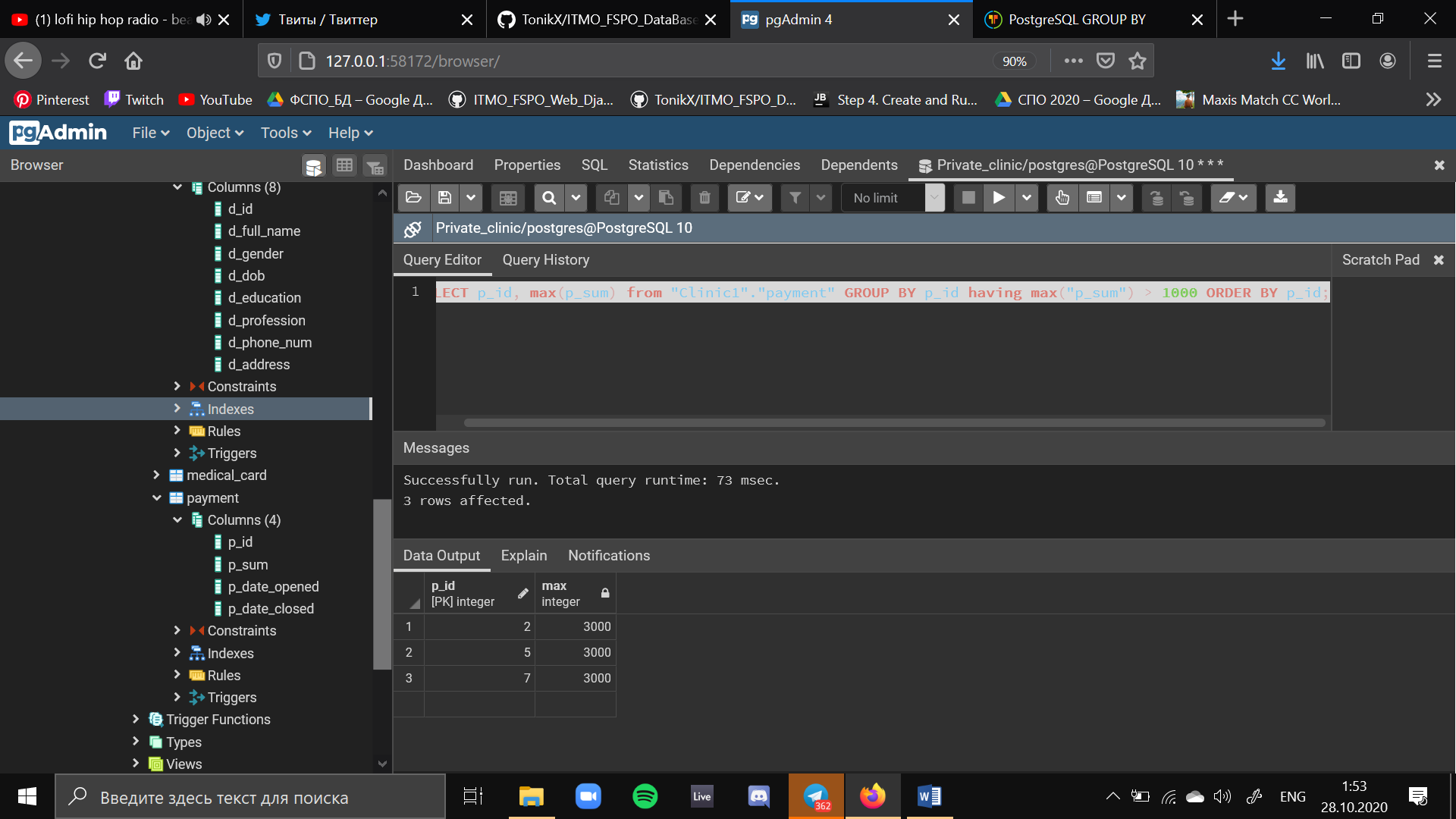
1. На рисунке 7 предоставлена информация о максимальной сумме оплаты за прием.

SELECT max("payment"."p\_sum") as Max\_sum from "clinic"."payment" join "clinic"."appointment" on "appointment"."a\_payment" = "payment"."p\_id";

  
*Рисунок 7 - Результат запроса*

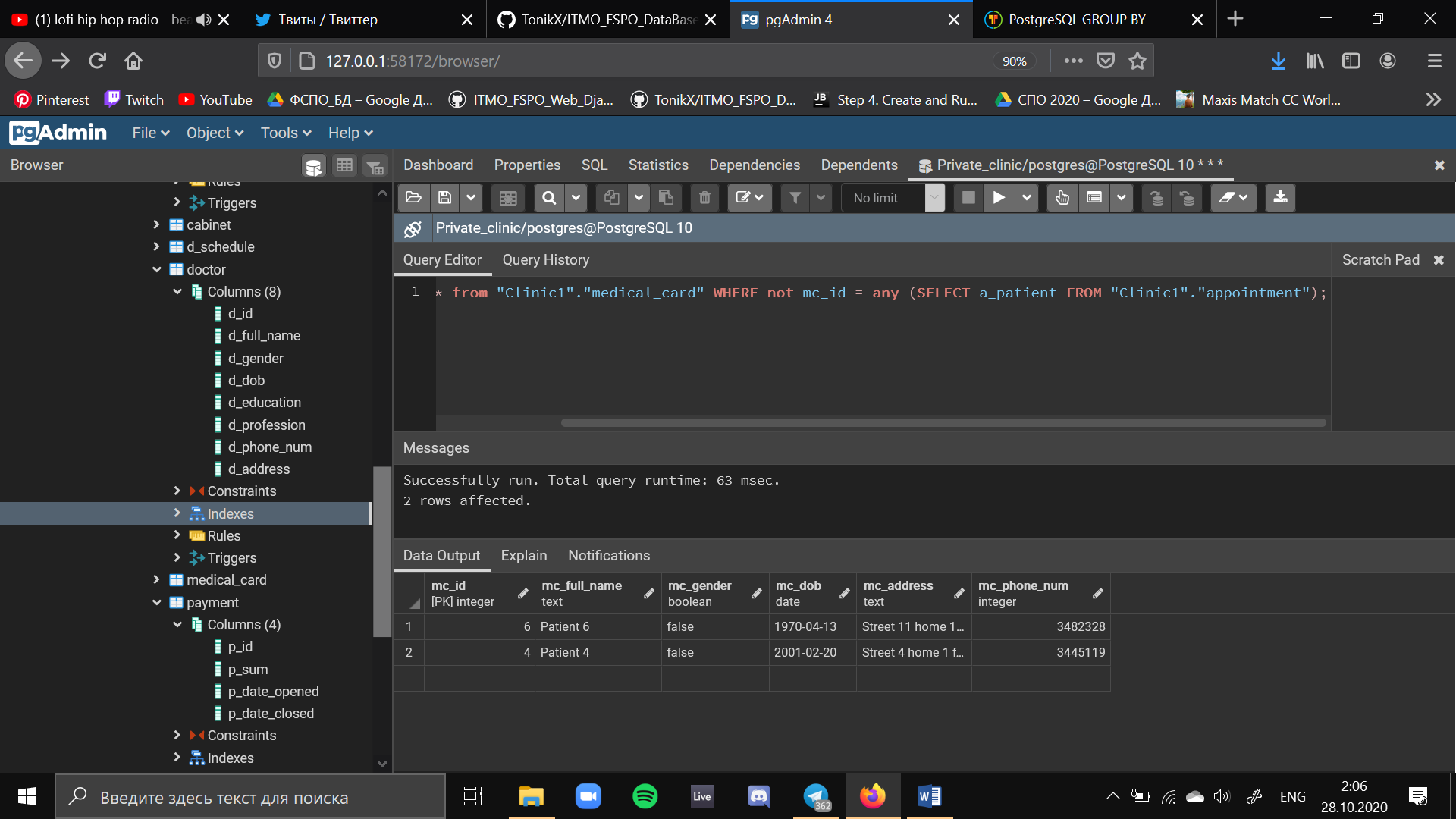
1. Вывод номеров счетов, за которые должны были заплатить больше, чем 1000 , организованная по номеру счетов (рисунок 8).

SELECT p\_id, max(p\_sum) from "clinic"."payment" GROUP BY p\_id having max("p\_sum") > 1000 ORDER BY p\_id;

*Рисунок 8 - Результат запроса*

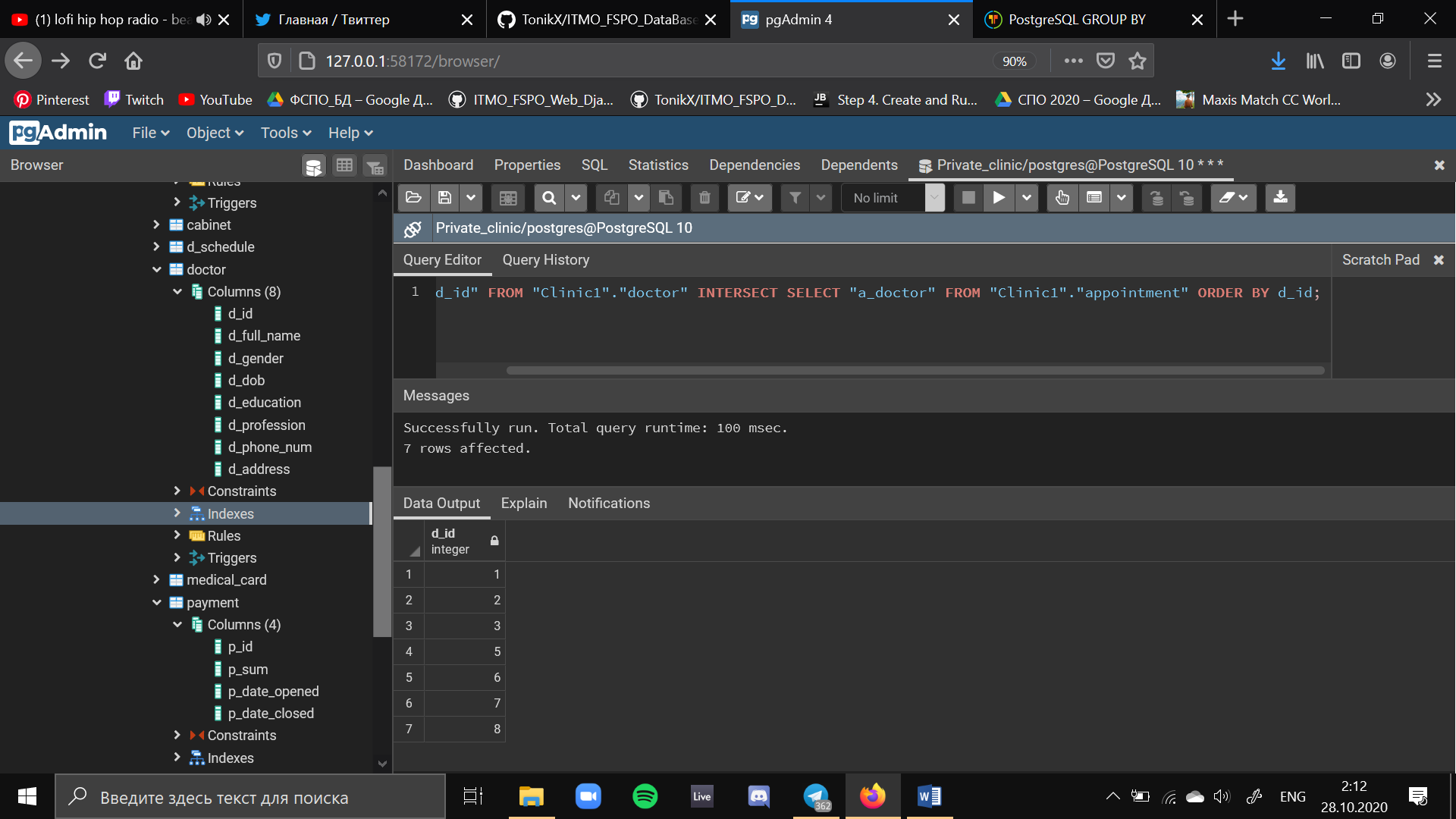
1. На рисунке 9 представлена информация о пациентах, которые зарегистрированы, но ещё никогда не были на приёме.

SELECT distinct \* from "clinic"."medical\_card" WHERE not mc\_id = any (SELECT a\_patient FROM "clinic"."appointment");

*Рисунок 9 - Результат запроса*

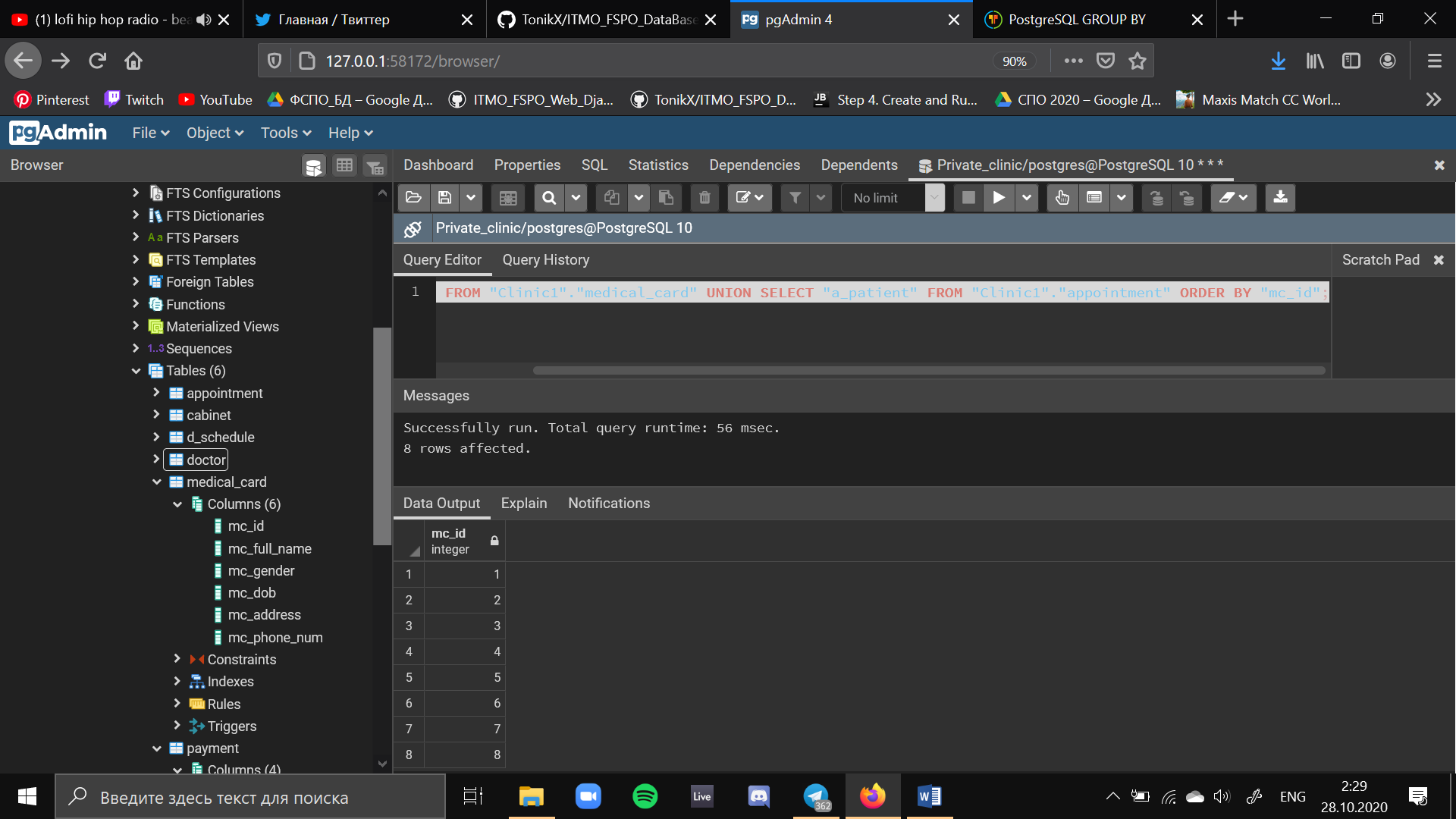
1. Нарисунке 10 представлена информация об врачах, которые уже принимали пациентов, организованно по возрастанию ID врача.

SELECT "d\_id" FROM "clinic"."doctor" INTERSECT SELECT "a\_doctor" FROM "clinic"."appointment" ORDER BY d\_id;

*Рисунок 10 - Результат запроса*

1. На рисунке 11 представлена информация пациентах, которые есть и в таблице “мед\_карта ”, и в таблице “прием”.

SELECT mc\_id FROM "clinic"."medical\_card" UNION SELECT "a\_patient" FROM "clinic"."appointment" ORDER BY "mc\_id";

  
*Рисунок 11 - Результат запроса*

**Вывод:** в ходе выполнения лабораторной работы №5 были получены практические навыки создания SQL-запросов.