МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Отчет по лабораторной работе №1

«Хранимые процедуры»

по дисциплине «Базы данных»

Вариант 6

Выполнил студент группы 220681

Хохряков Д.А.

Проверил:

к.т.н., доц. Афанасьева С.М.

Тула 2020

# **ЦЕЛЬ РАБОТЫ**

Целью лабораторной работы является изучение и практическое применение хранимых процедур.

# **ЗАДАНИЕ НА РАБОТУ**

1. Создайте функцию, возвращающую количество записей в таблице по указанному значению внешнего ключа. Входной параметр - значение ключа.
2. Создайте процедуру, изменяющую поле строки. Входные параметры - первичный ключ строки и новое значение изменяемого поля.
3. Создайте новую таблицу TEMP для занесения временных данных. Напишите процедуру (используя явный курсор) для выборки первых пяти записей какой-либо таблицы, соответствующих определенному критерию, и занесения их в таблицу TEMP.
4. Приведите:

* смысл процедуры или функции на языке, понятном пользователю в данной предметной области.
* оператор, создающий данную процедуру или функцию
* состояние данных до выполнения процедуры или функции
* оператор, запускающий процедуру или функцию
* состояние данных после выполнения процедуры или функции

# **ХОД РАБОТЫ**

1. Сначала создадим функцию для подсчёта количества почтовых отделений, в которых есть газеты в определённом количестве.

Для создания функции нужно использовать команду CREATE OR REPLACE FUNCTION, после которой идёт имя, параметры (countof), возвращаемый тип и блок кода. В коде с помощью команды SELECT подсчитывается количество строк, у которых значение столбца count совпадает со значение параметра countof. Полученное число возвращается командой RETURN (Рисунок 1).

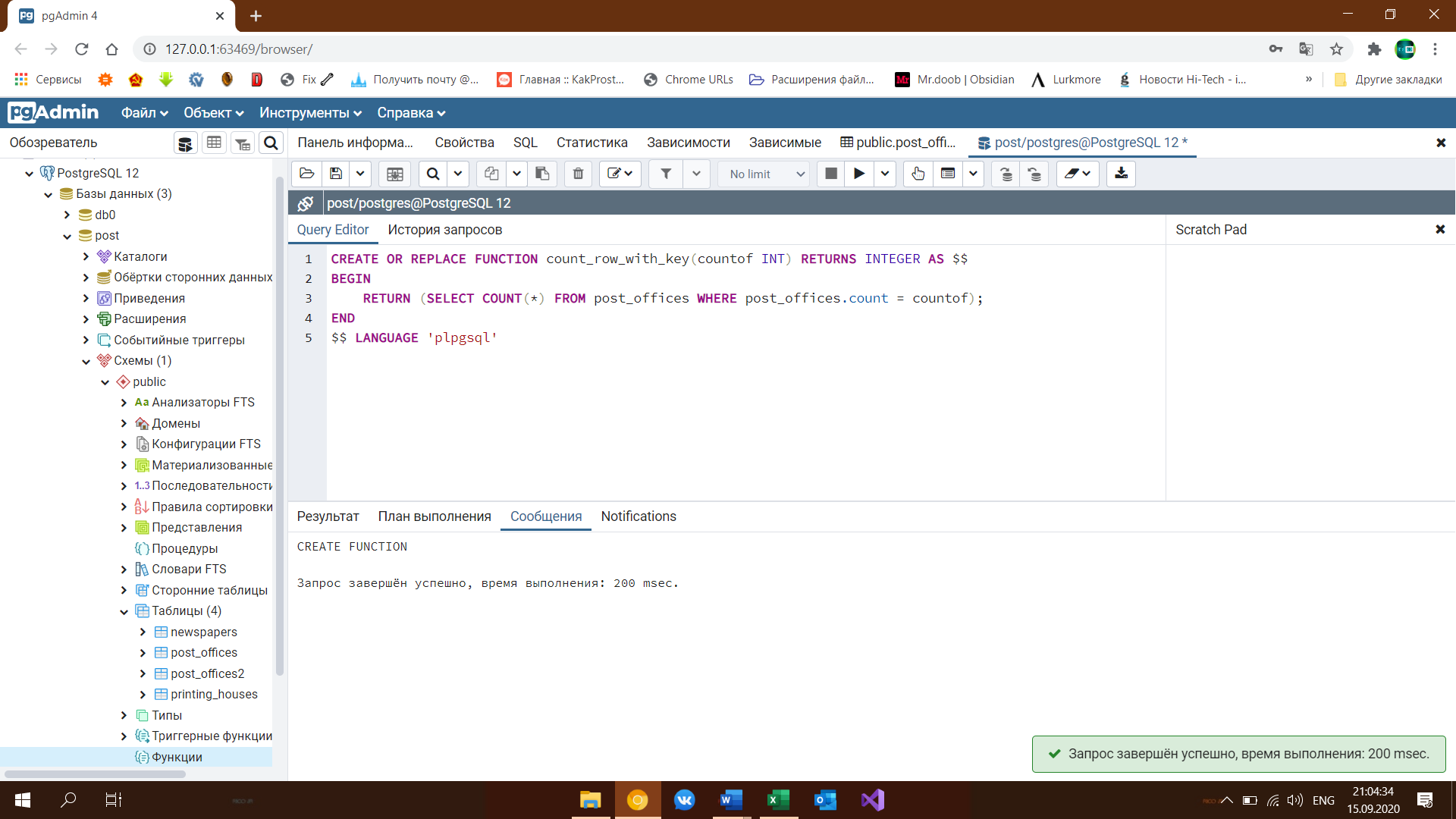


Рисунок 1 – Создание функции count\_row\_with\_key

Подсчёт строк будет производиться в таблице post\_offices, содержимое которой представлено на рисунке 2.

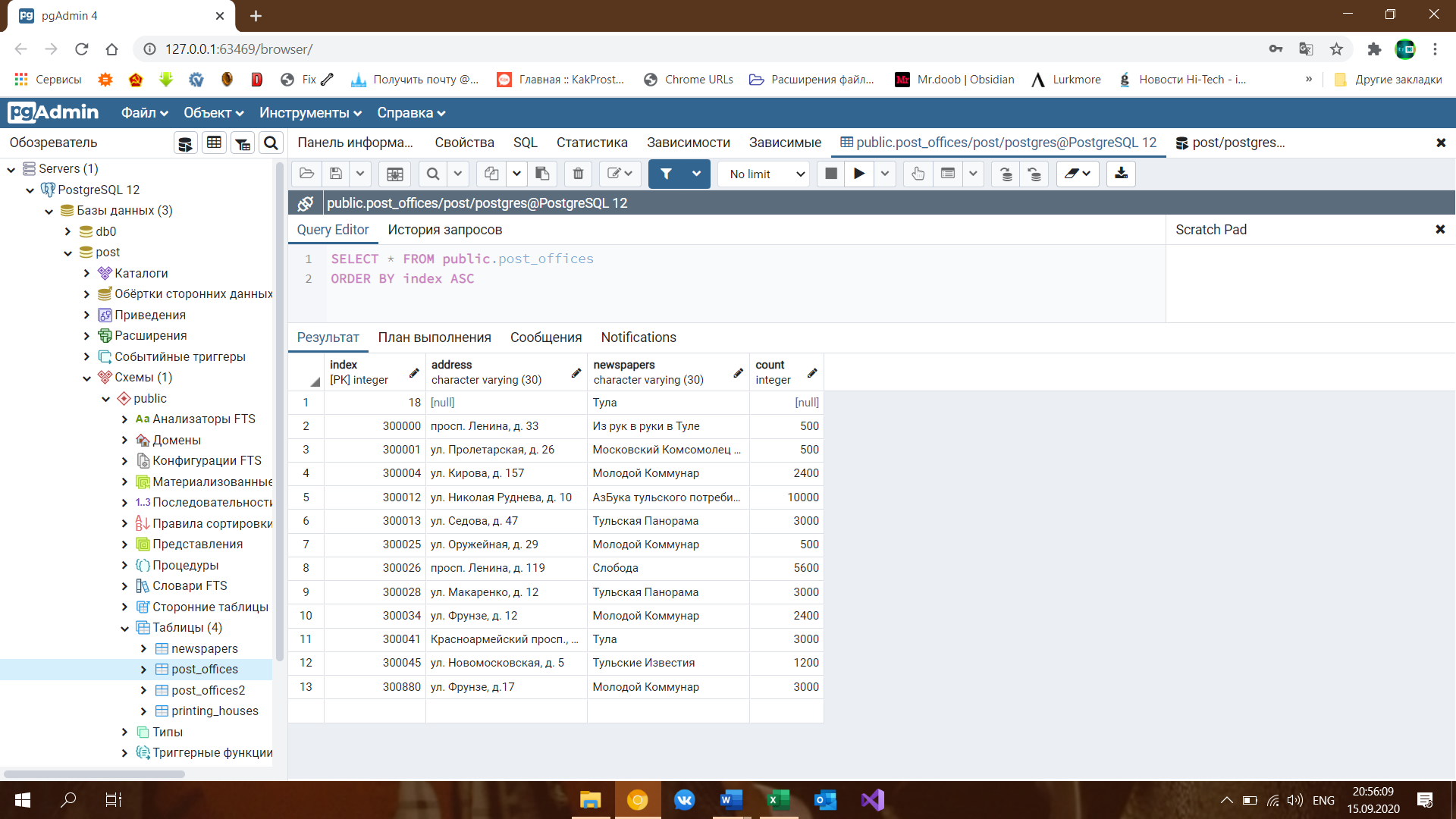


Рисунок 2 – Содержание таблицы post\_offices

Чтобы сразу выполнить код на языке plpgsql нужно использовать команду DO, после которой идёт блок кода. В этом коде для проверки вызывается ранее созданная функция count\_row\_with\_key, куда в качестве параметра передаётся 1. Для удобства значение, возвращённое функцией, будет выведено с помощью команды RAISE NOTICE (Рисунок 3).

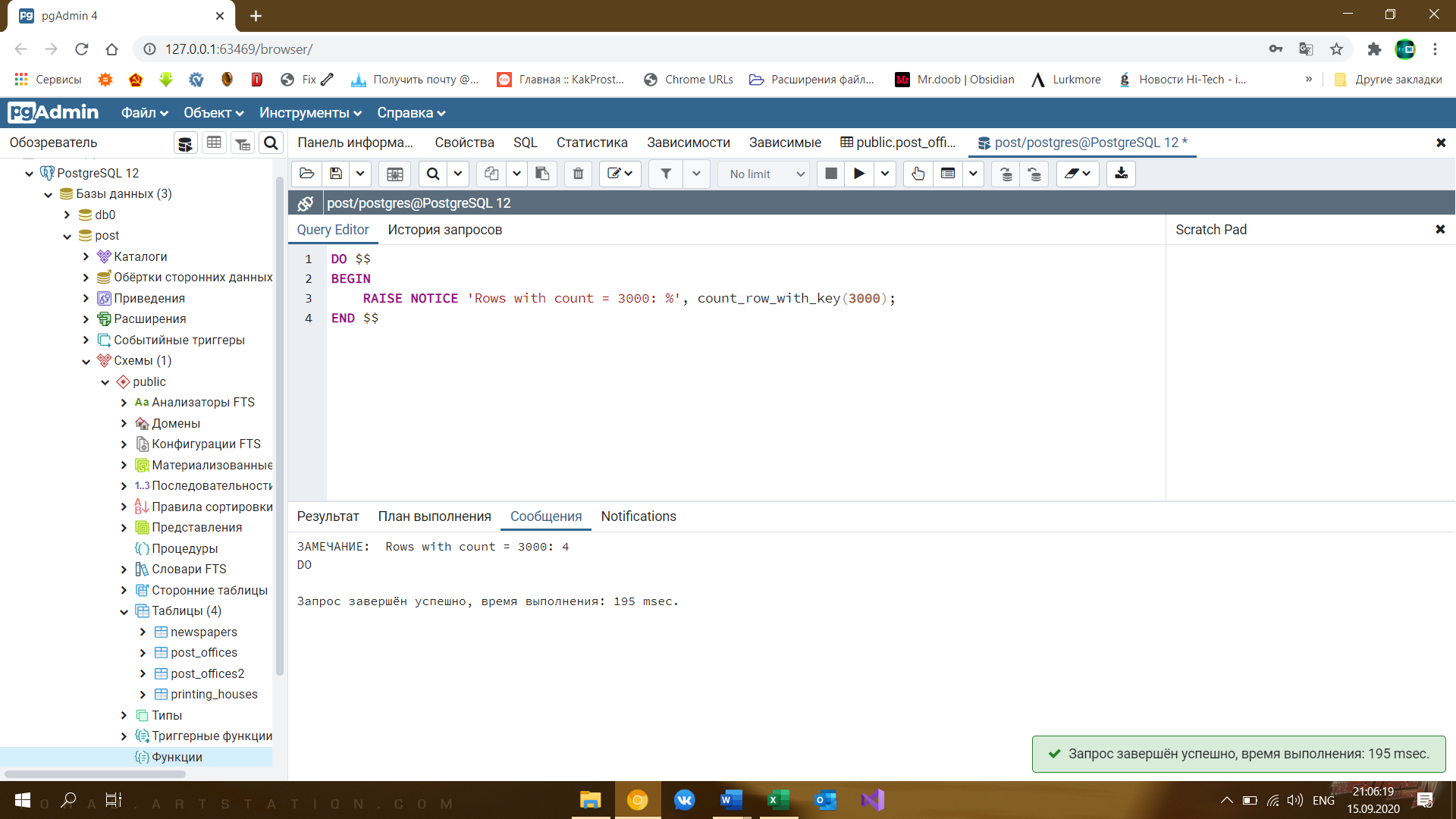


Рисунок 3 – Запуск функции count\_row\_with\_key

1. Теперь создадим процедуру, которая меняет у заданного запроса клиента требуемую черту.

Для создания процедуры нужно использовать команду CREATE OR REPLACE PROCEDURE, после которой идёт имя, параметры (newspaper и countof) и блок кода. В коде с помощью команды UPDATE устанавливается значение параметра countof для поля count в тех строках, у которых поле newspapers совпадает с параметром newspaper (Рисунок 4).

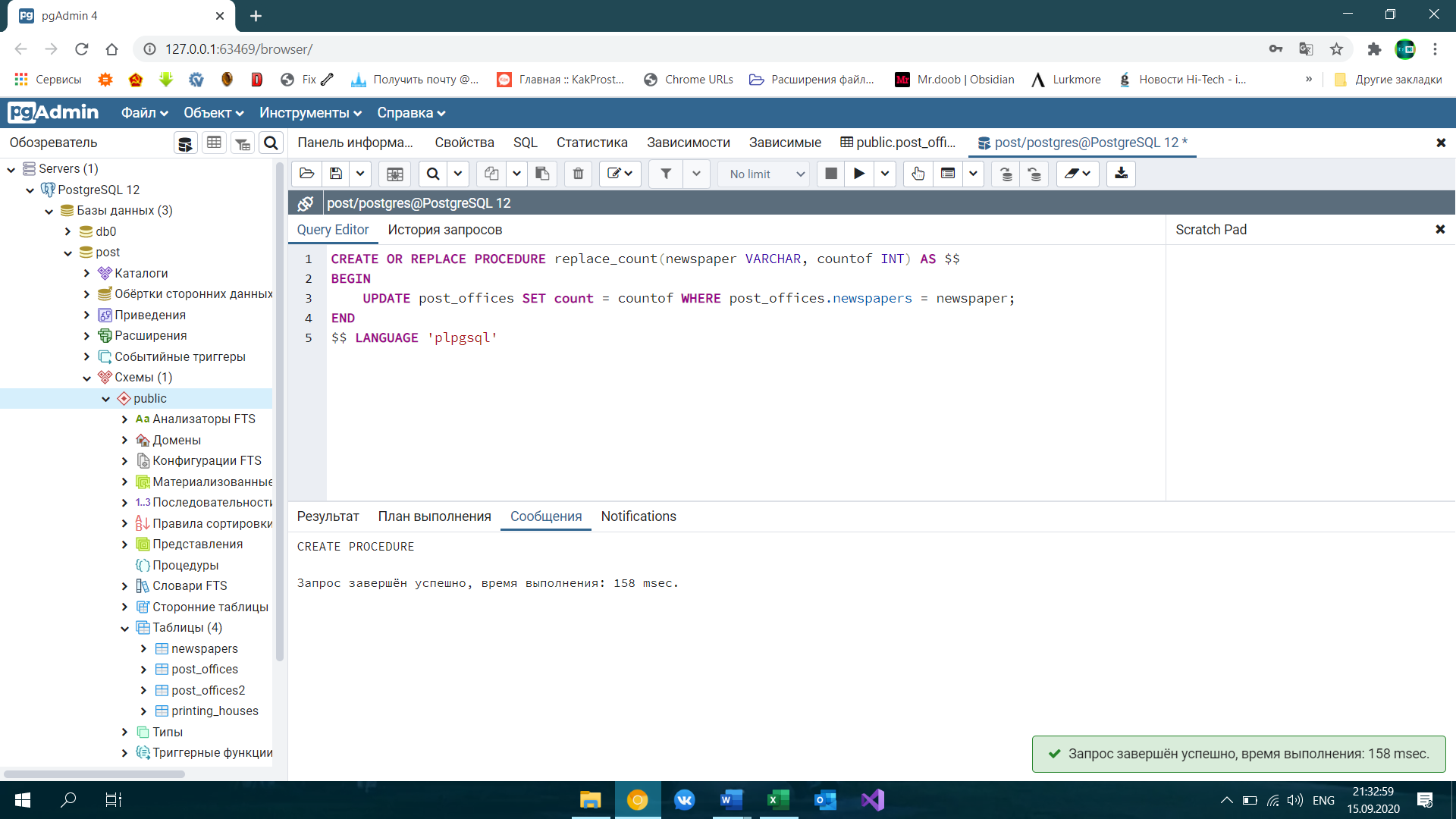


Рисунок 4 – Создание процедуры replace\_count

Данная процедура будет применяться к таблице post\_offices, содержимое которой представлено на рисунке 5.

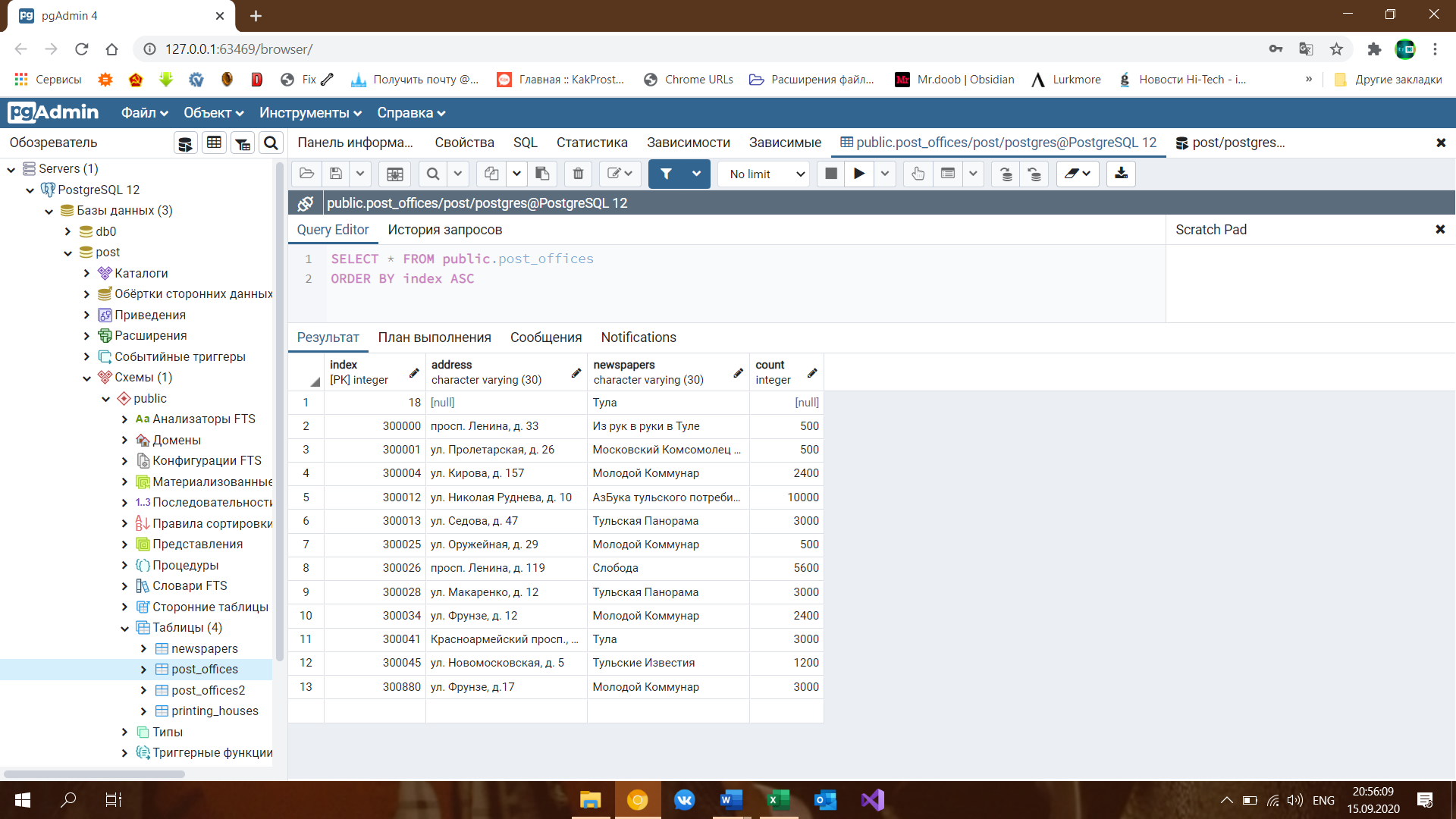


Рисунок 5 – Содержание таблицы post\_offices

Чтобы вызвать процедуру, нужно использовать команду CALL. При запуске процедуры в неё были переданы значения параметров ‘Молодой Коммунар’ и 1000 (Рисунок 6).

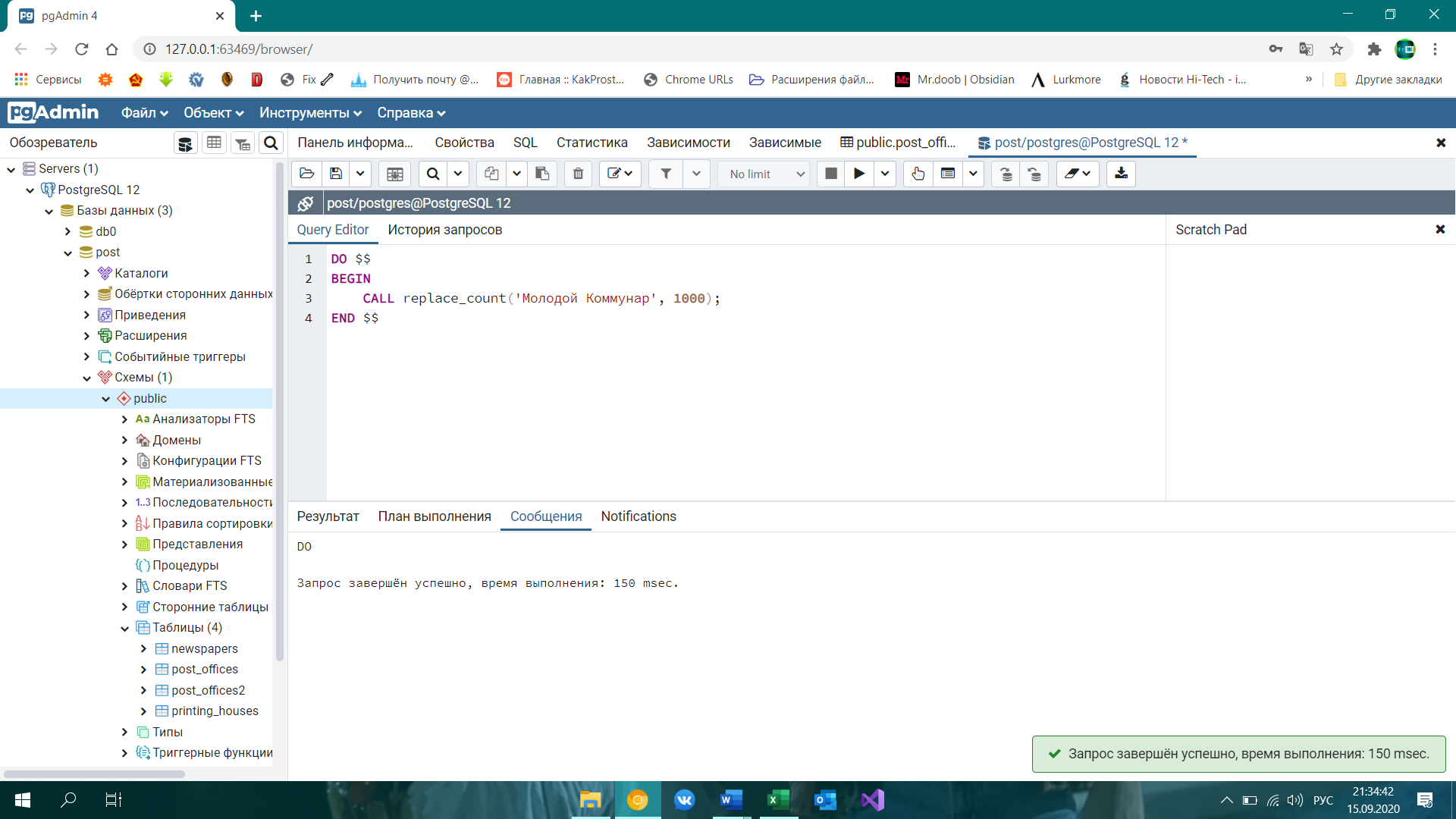


Рисунок 6 – Применение процедуры replace\_count

После применения процедуры replace\_count значения поля count в строках с newspapers = ‘Молодой Коммунар’ стало равным 1000 (Рисунок 7).

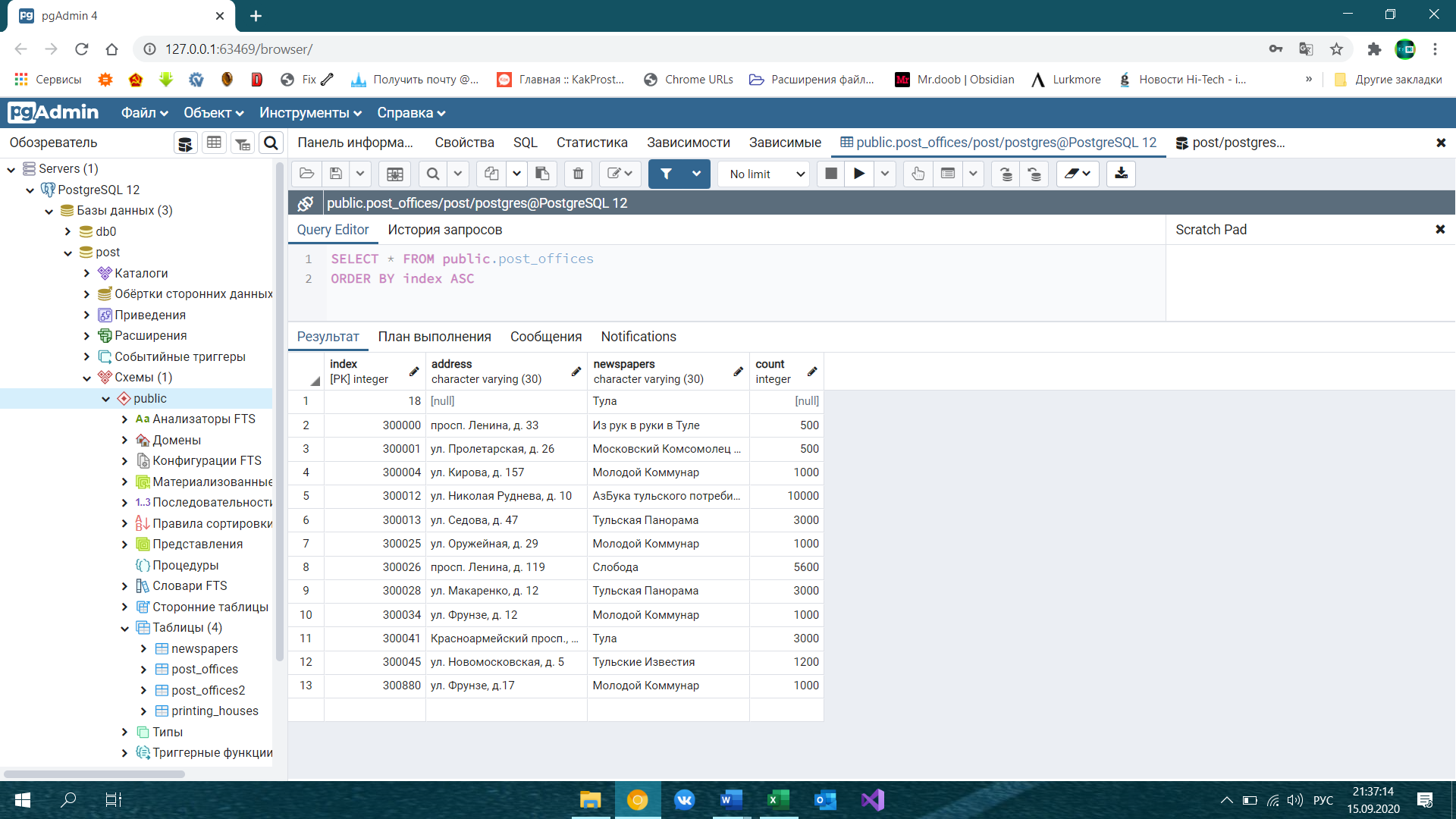


Рисунок 7 – Содержание таблицы post\_offices после применения процедуры

1. Далее создадим процедуру для занесения в отдельную таблицу 5 почтовых отделений, у которых индекс больше заданного.

Для создания процедуры нужно использовать команду CREATE OR REPLACE PROCEDURE, после которой идёт имя, параметры (from\_index) и блок кода. В коде сначала объявляются переменные index, post\_office и cur, которая является явным курсором (так как объявляется в поле DECLARE) для запроса SELECT. Этот запрос получает из таблицы post\_offices 5 строк, в которых значения поля index больше значения параметра from\_index. Далее открывается курсор и с помощью цикла все строки из него заносятся в таблицу temp командой INSERT. После цикла курсор закрывается (Рисунок 8).

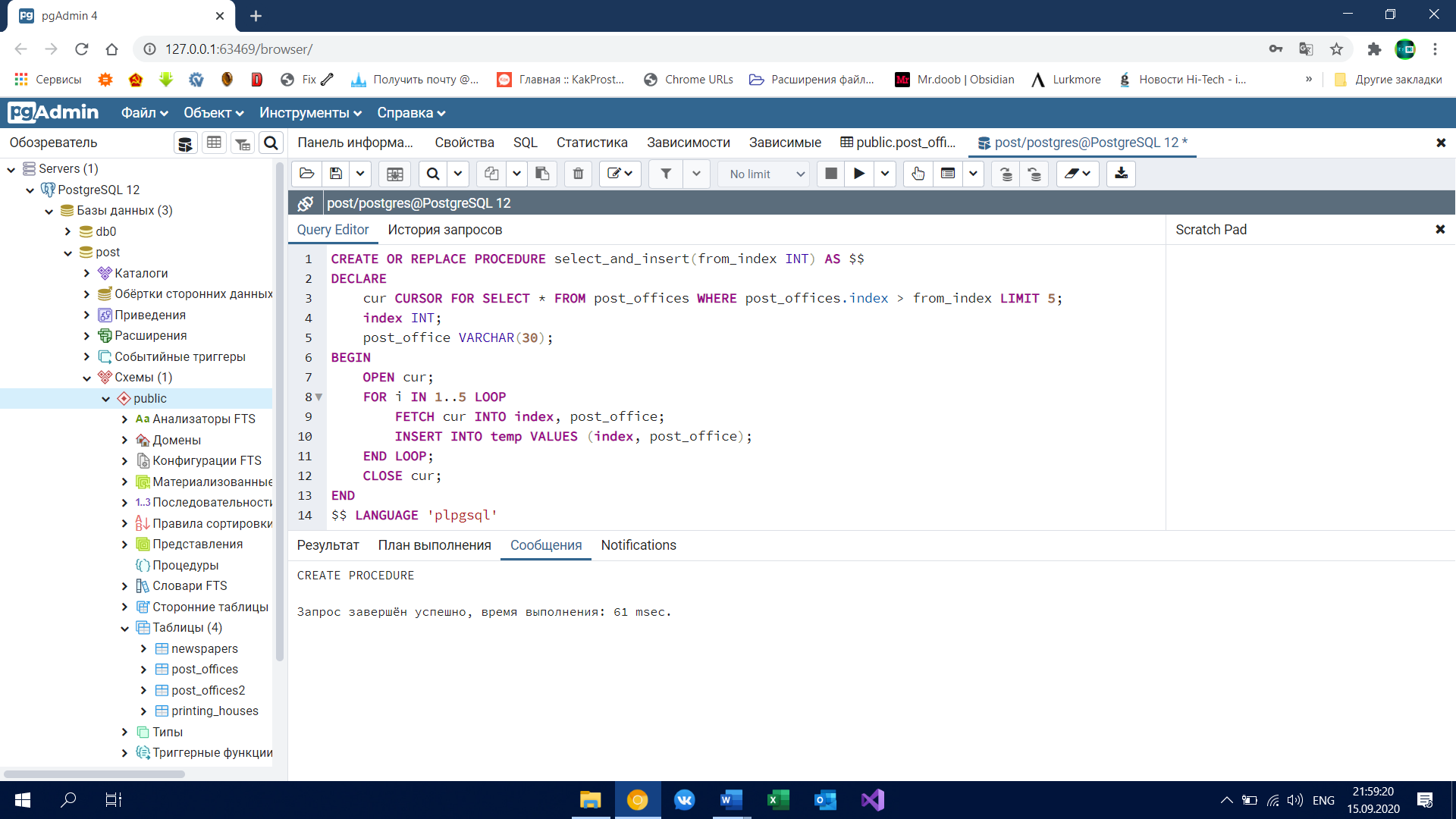


Рисунок 8 – Создание процедуры select\_and\_insert

Процедура будет применяться к таблице post\_offices, содержимое которой представлено на рисунке 9.

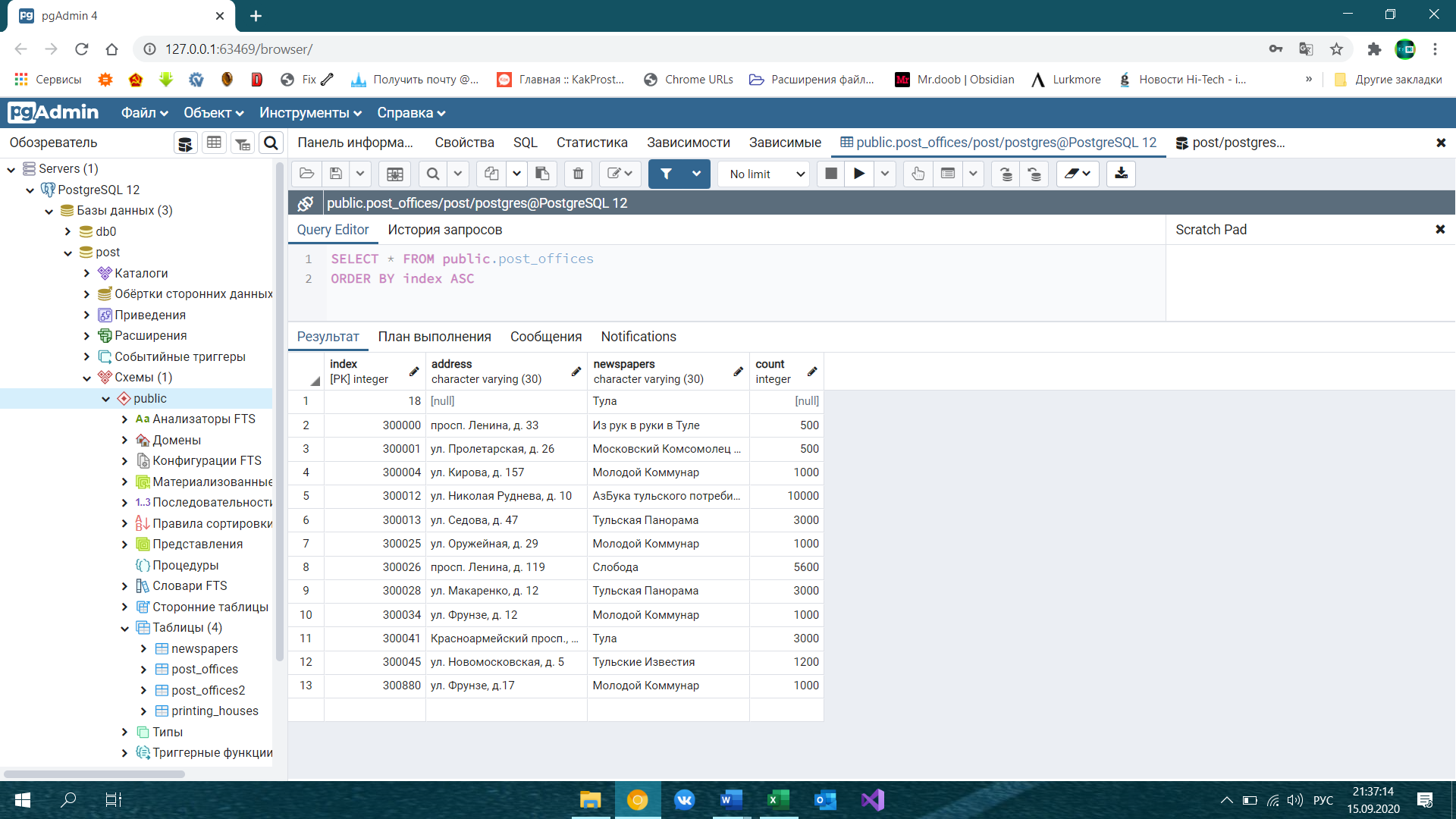


Рисунок 9 – Содержание таблицы post\_offices

Чтобы вызвать процедуру, нужно использовать команду CALL. При запуске процедуры в неё было передано значение 300012 (Рисунок 10).

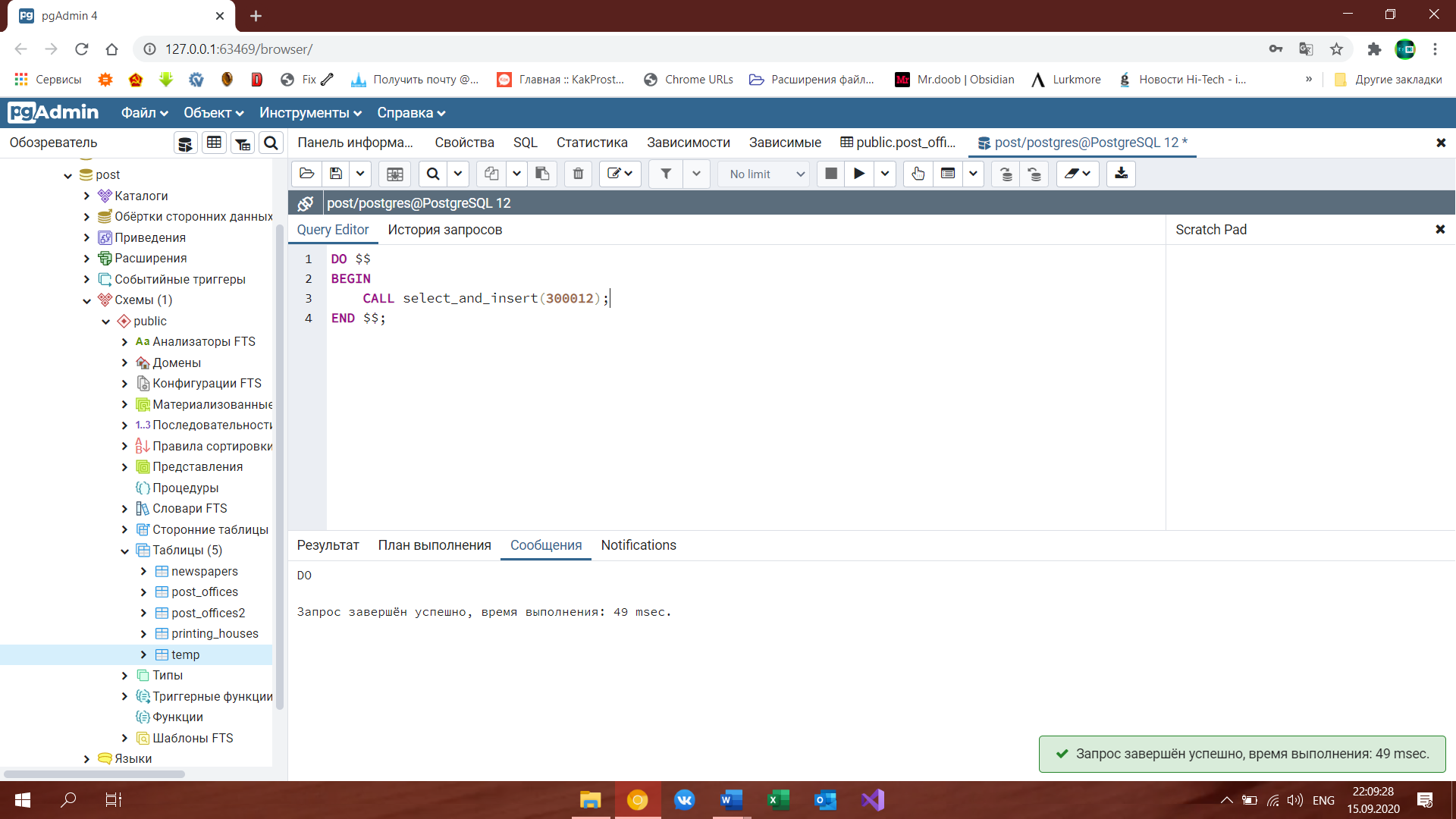


Рисунок 10 – Применение процедуры select\_and\_insert

После применения процедуры select\_and\_insert в таблице temp появилось пять новых строк (Рисунок 7).

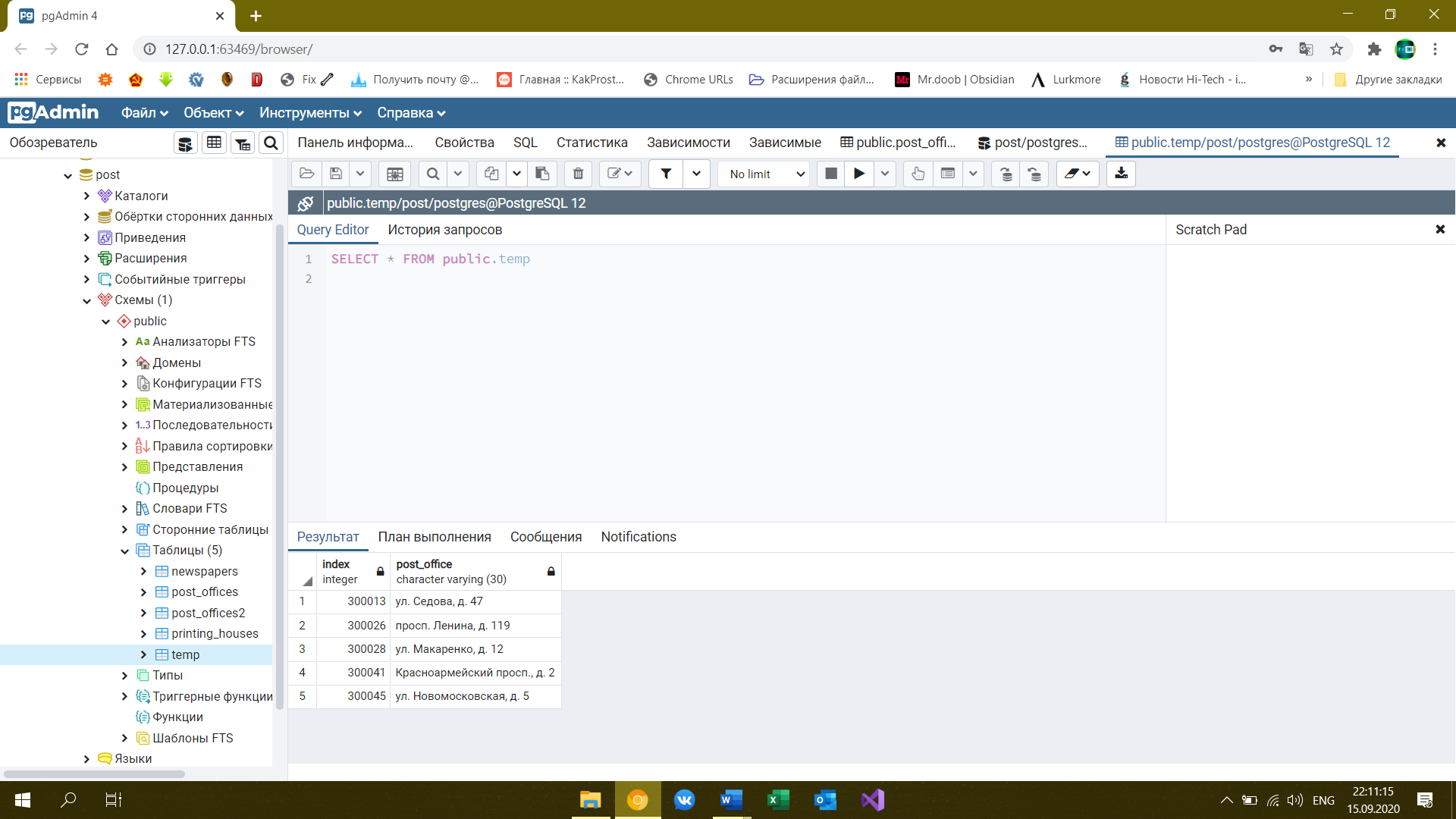


Рисунок 11 – Содержание таблицы temp после применения процедуры

# **ВЫВОД**

В ходе данной лабораторной работы были изучены и практически применены хранимые процедуры.