ОТЧЕТ

по лабораторной работе № 3

«Элементарные выборки посредством оператора SELECT. Агрегатные функции SQL»

по дисциплине

«ТЕОРИЯ БАЗ ДАННЫХ»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверила:

Лебедева М.А.

* 1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является изучение работы оператора SELECT языка SQL, а также изучение возможностей обработки данных с помощью агрегатных функций языка SQL.

* 1. ЗАДАНИЕ НА РАБОТУ

Изучить методические рекомендации по выполнению работы.

Открыть в phpMyAdmin созданную в л/р №1и №2 базу данных, содержащую одну таблицу. Обязательно убедиться в наличии в таблице записей необходимых для реализации запросов на выборку. В случае отсутствия таковых, добавить необходимые записи в таблицу.

Продемонстрировать все записи таблицы, указанной в варианте задания, с помощью SQL-запроса.

С помощью SQL-запроса продемонстрировать действие модификатора DISTINCT.

Ограничить вывод данных с помощью SQL-запроса, используя WHERE с простым условием.

Ограничить вывод данных с помощью SQL-запроса, используя WHERE и составное условие.

С помощью SQL-запроса продемонстрировать действие специальных функций IN, BETWEEN, LIKE и IS NULL в условии.

С помощью SQL-запроса продемонстрировать работу специальных функций с условием NOT.

Ознакомиться с принципами работы агрегатных функций COUNT, SUM, AVG, MAX, MIN. Составить SQL-запрос с одной из агрегатных функций.

С помощью SQL-запроса продемонстрировать использование COUNT(\*).

С помощью SQL-запроса продемонстрировать выполнение простых вычислений в запросе.

Использовать простое вычисление, как параметр агрегатной функции, в SQL-запросе.

Ознакомиться с использованием предложения GROUP BY, продемонстрировать его работу с помощью SQL-запроса.

Ознакомиться с использованием предложения HAVING, продемонстрировать его работу с помощью SQL-запроса.

С помощью SQL-запросов выполнить следующие задания по варианту (**вариант № 15**):

осуществить фильтрацию информации по полю «Адрес»;

получить результат агрегатной функции «Максимальная площадь» по полю группировки «Номер фирмы»;

выполнить сортировку информации по убыванию значений в поле, выбранном пользователем из списка всех полей таблицы (направление сортировки задано вариантом задания).

Заданная вариантом № 15 предметная область:

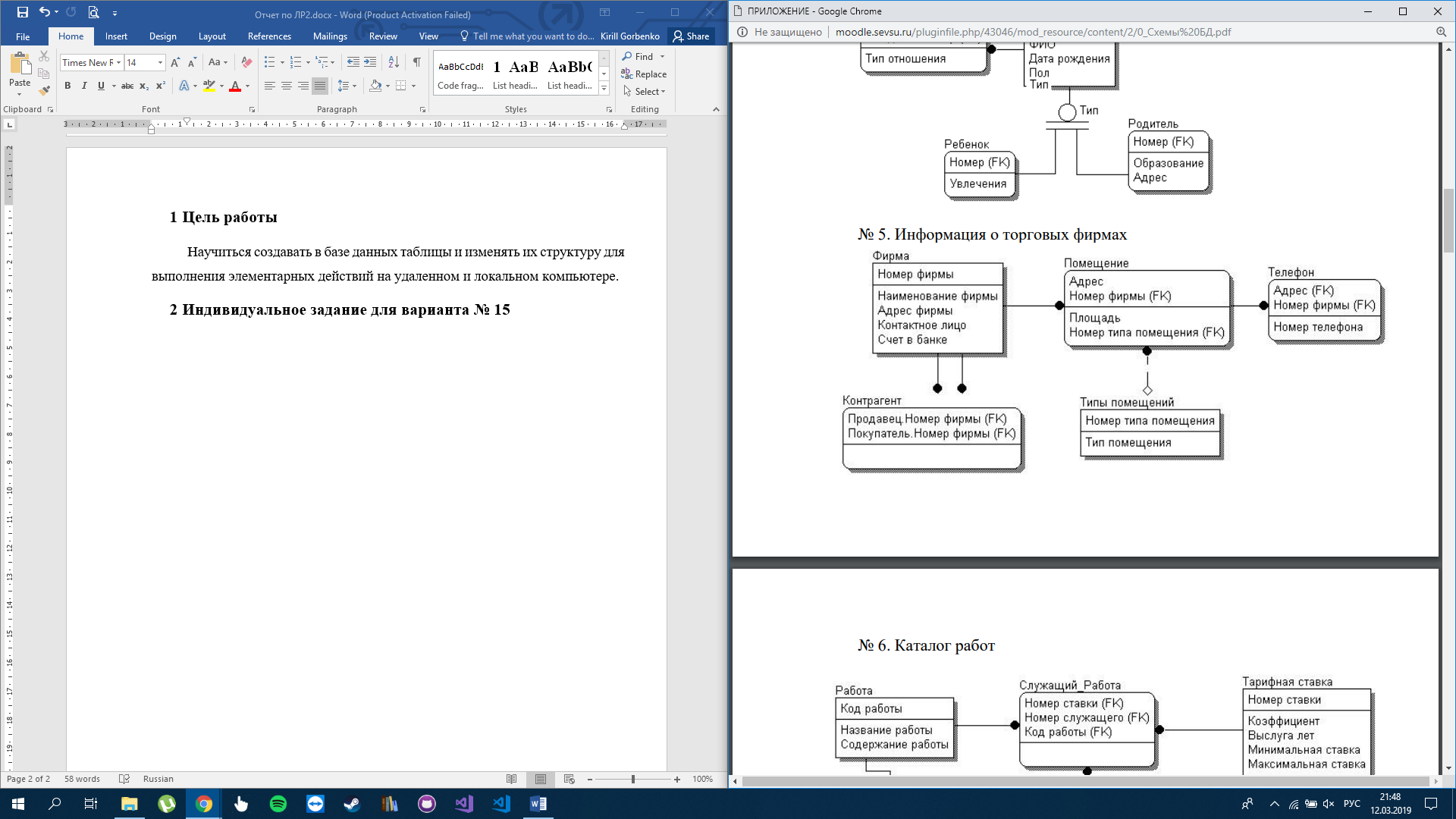


Рисунок № 1 – Схема № 5 «Информация о торговых фирмах»

* 1. ХОД РАБОТЫ
     1. Выборка всех данных таблицы

Введем следующий SQL запрос для выборки всех данных таблицы:

SELECT \* FROM room

Результат запроса представлен на рисунке № 2:

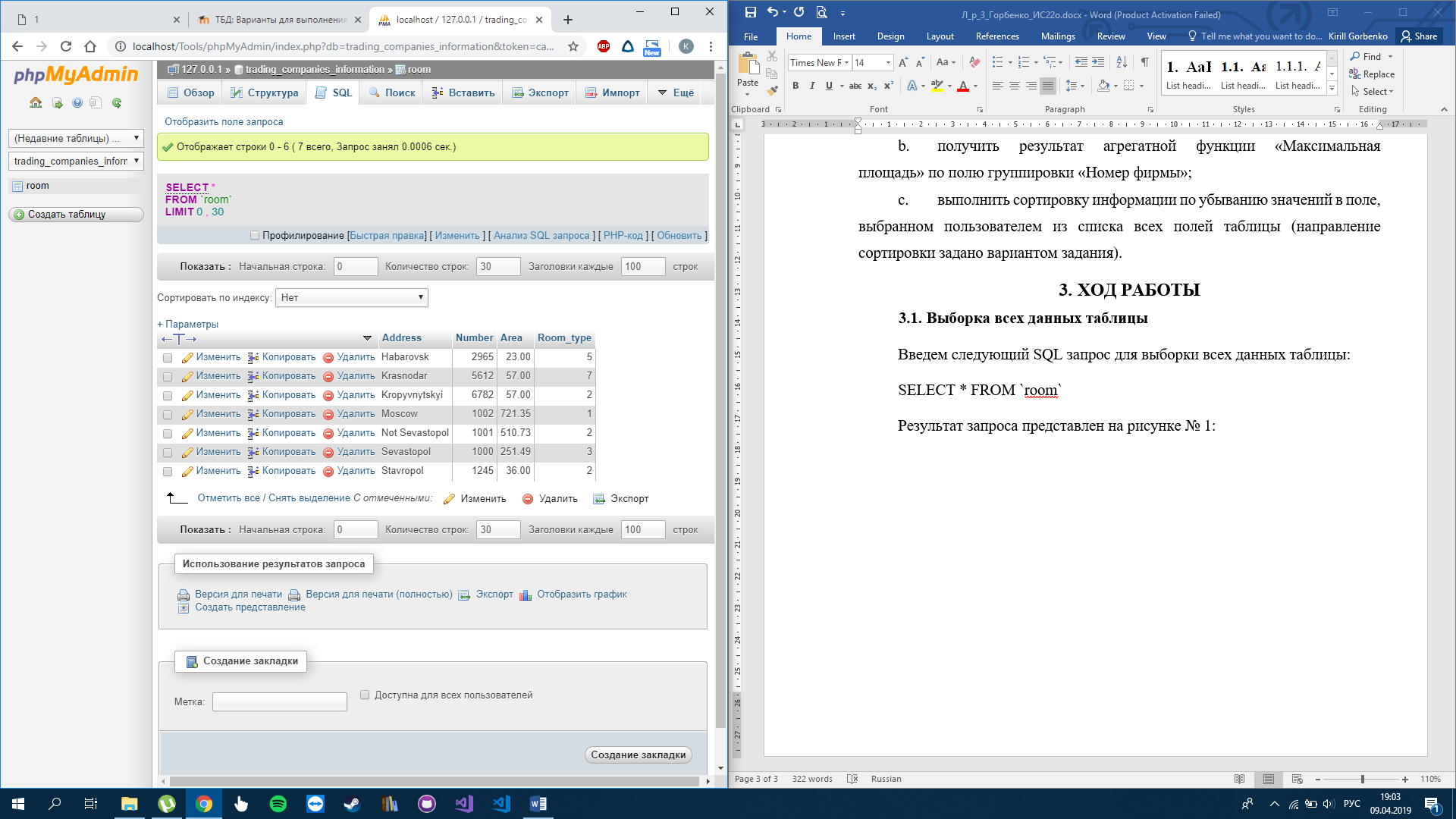


Рисунок № 2 – Результат запроса на выборку всех данных

* + 1. Выборка записей с использованием модификатора DISTINCT

Напишем SQL запрос на выборку из таблицы «Помещение» всех уникальных значений поля «Площадь». Выбрано именно это поле, т.к. оно содержит записи с одинаковыми значениями. Текст запроса:

SELECT DISTINCT Area FROM room

Результат запроса изображен на рисунке № 3:

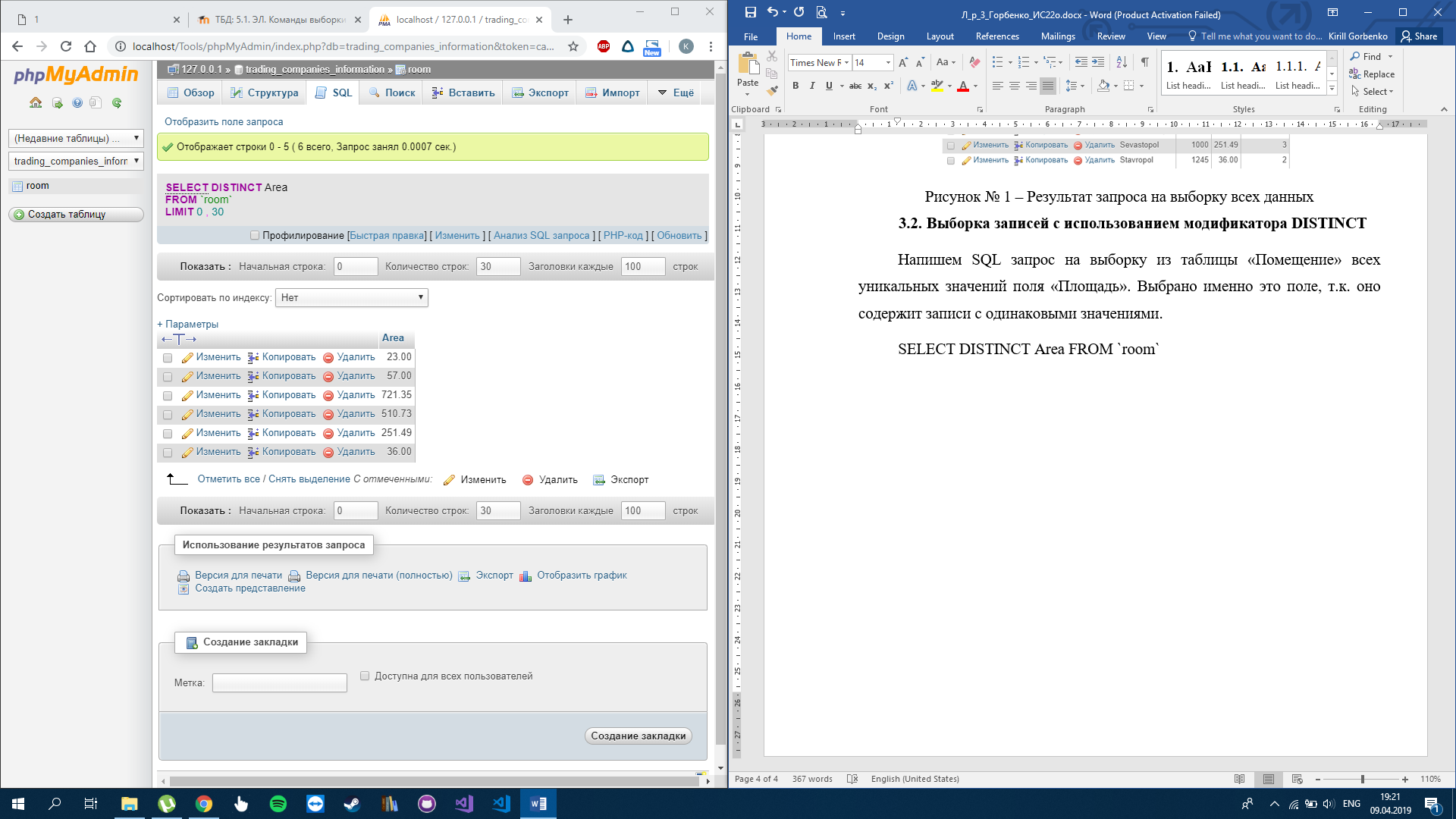


Рисунок № 3 – Результат запроса с использованием модификатора DISTINCT

* + 1. Выборка записей с использованием простых условий

Выберем все данные помещений, площадь которых больше 100.

SELECT \* FROM room WHERE Area > 100

Результат запроса изображен на рисунке № 4:

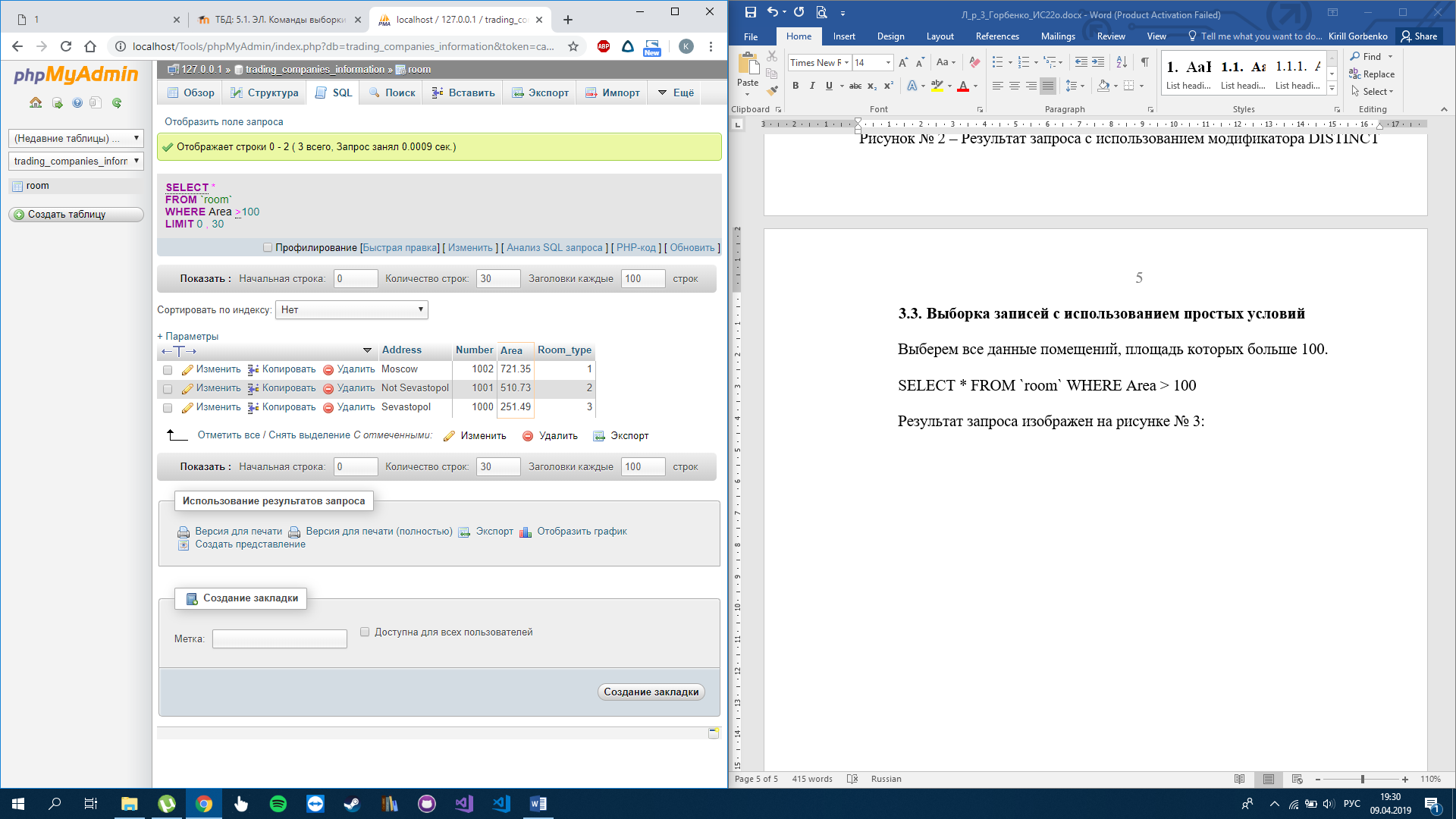


Рисунок № 4 – Результат запроса на выборку с простым условием

* + 1. Выборка записей с использованием составных условий

Выберем все данные помещений второго типа, площадь которых более 100. Используем следующий запрос:

SELECT \* FROM room WHERE Room\_type = 2 And Area > 100

Результат запроса представлен на рисунке № 5:

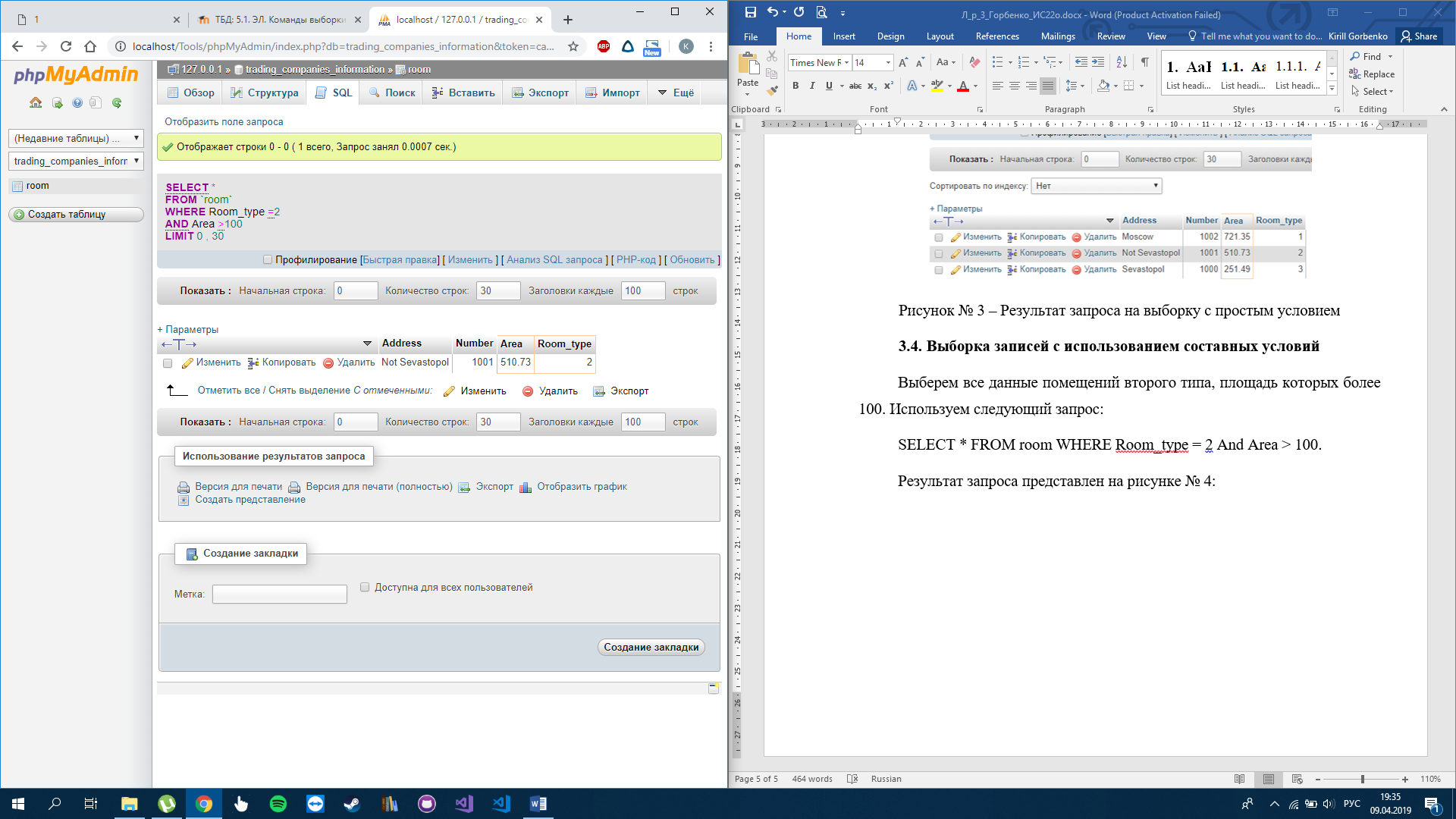


Рисунок № 5 – Результат запроса на выборку со сложным условием

* + 1. Выборка записей с использованием функций IN, BETWEEN, LIKE, IS NULL

Выберем все данные помещений, номер которых лежит в промежутке между 1000 и 2000. Используем следующий запрос:

SELECT \* FROM room WHERE Number BETWEEN 1000 AND 2000

Результат запроса представлен на рисунке № 6:



Рисунок № 6 – Результат запроса с использованием функции BETWEEN

* + 1. Выборка данных с использованием условия NOT

Выберем все данные помещений, тип комнаты которых не равен 1, 2 и 3. Для этого используем следующий запрос:

SELECT \* FROM room WHERE Room\_type NOT IN (1, 2, 3)

Результат выполнения запроса представлен на рисунке № 7:

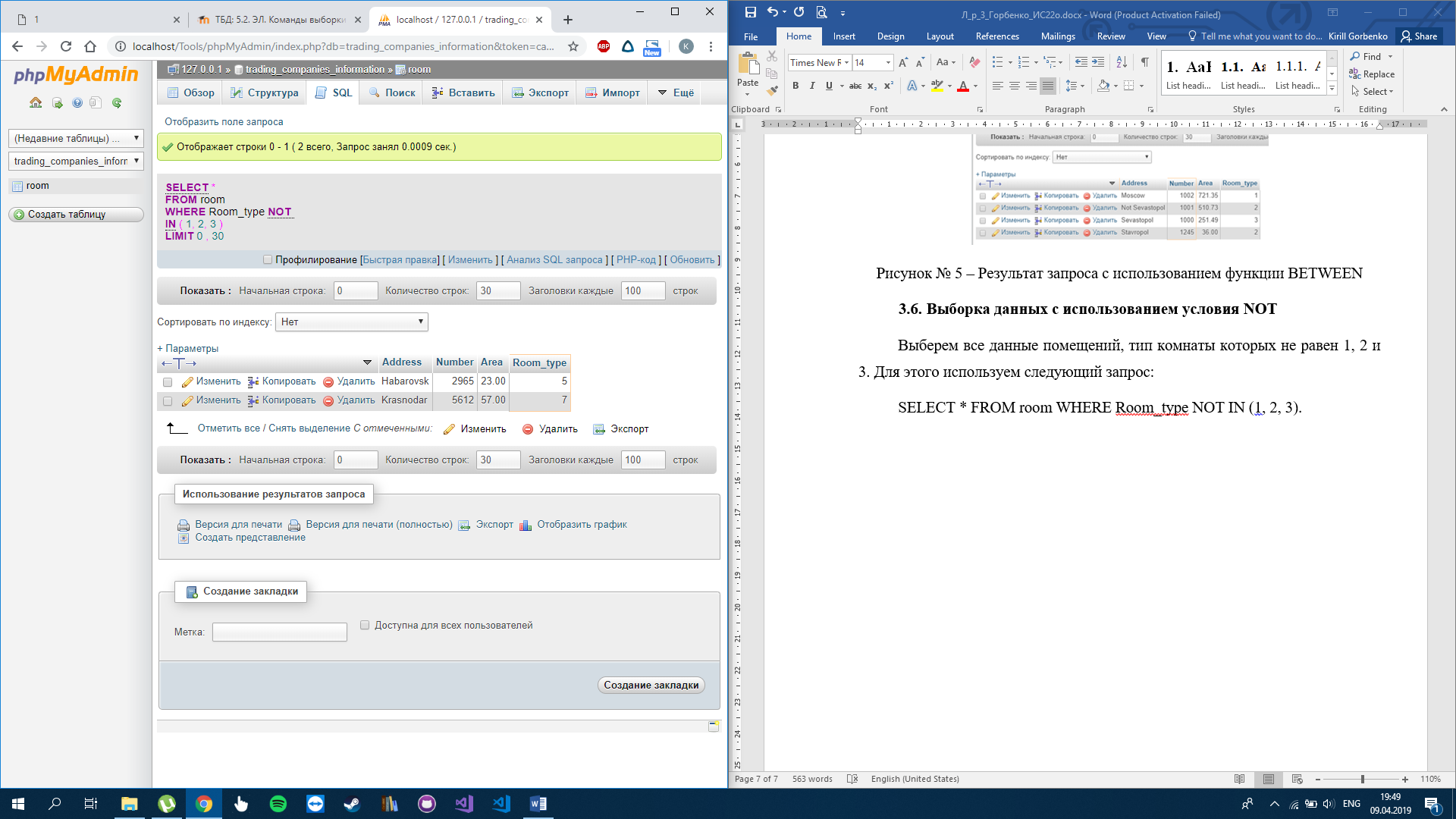


Рисунок № 7 – Результат запроса с использованием условия NOT

* + 1. Выборка данных и использование агрегатных функций

Выведем максимальную площадь помещения, используя следующий запрос:

SELECT MAX(Area) AS Max\_area FROM room

Результат выполнения запроса представлен на рисунке № 8:

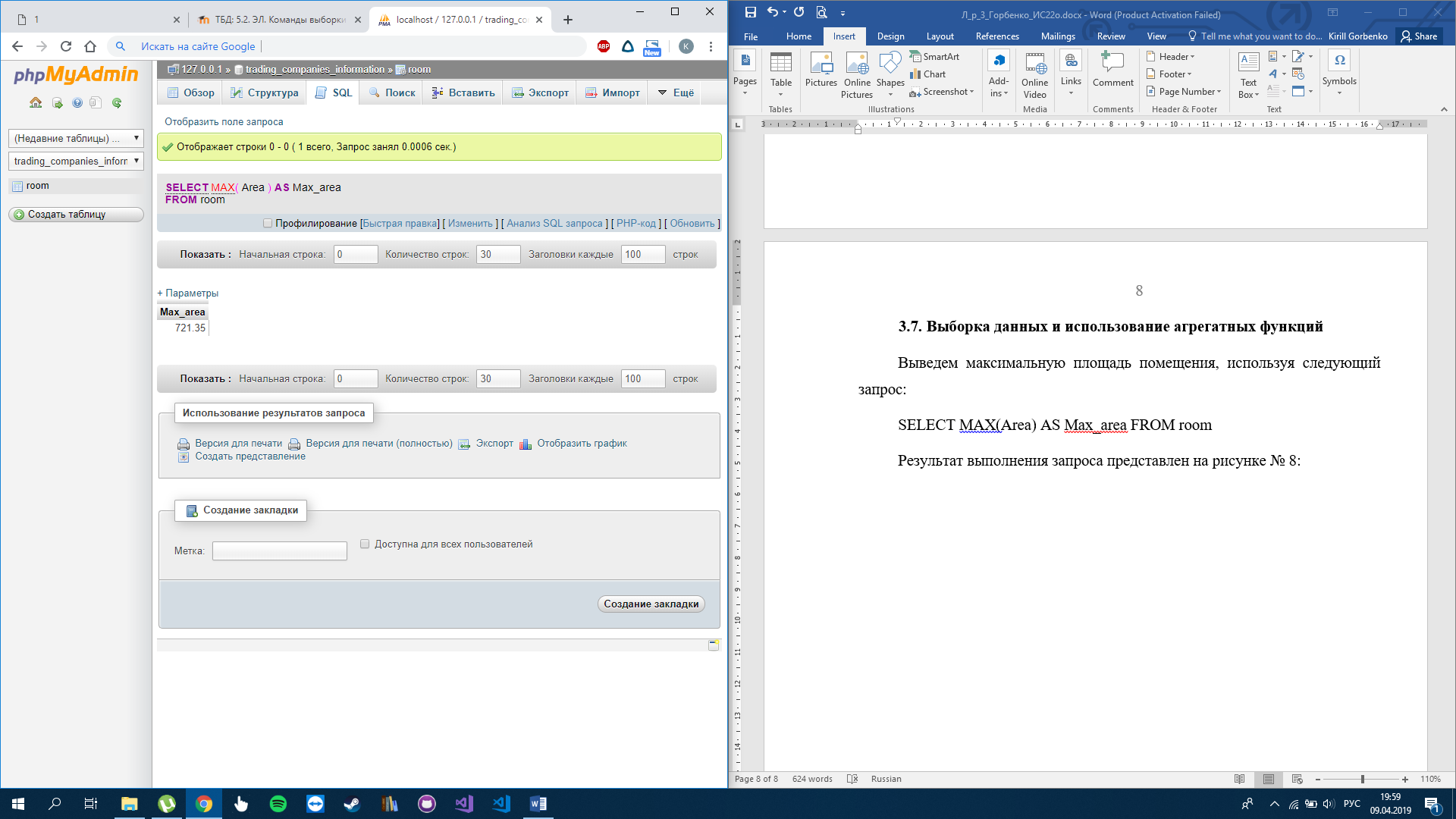


Рисунок № 8 – Результат выполнения запроса с использованием агрегатной функции

* + 1. Использование функции COUNT(\*)

Выведем количество записей в таблице с помощью следующего запроса:

SELECT COUNT(\*) AS amount FROM room

Результат приведен на рисунке № 9:

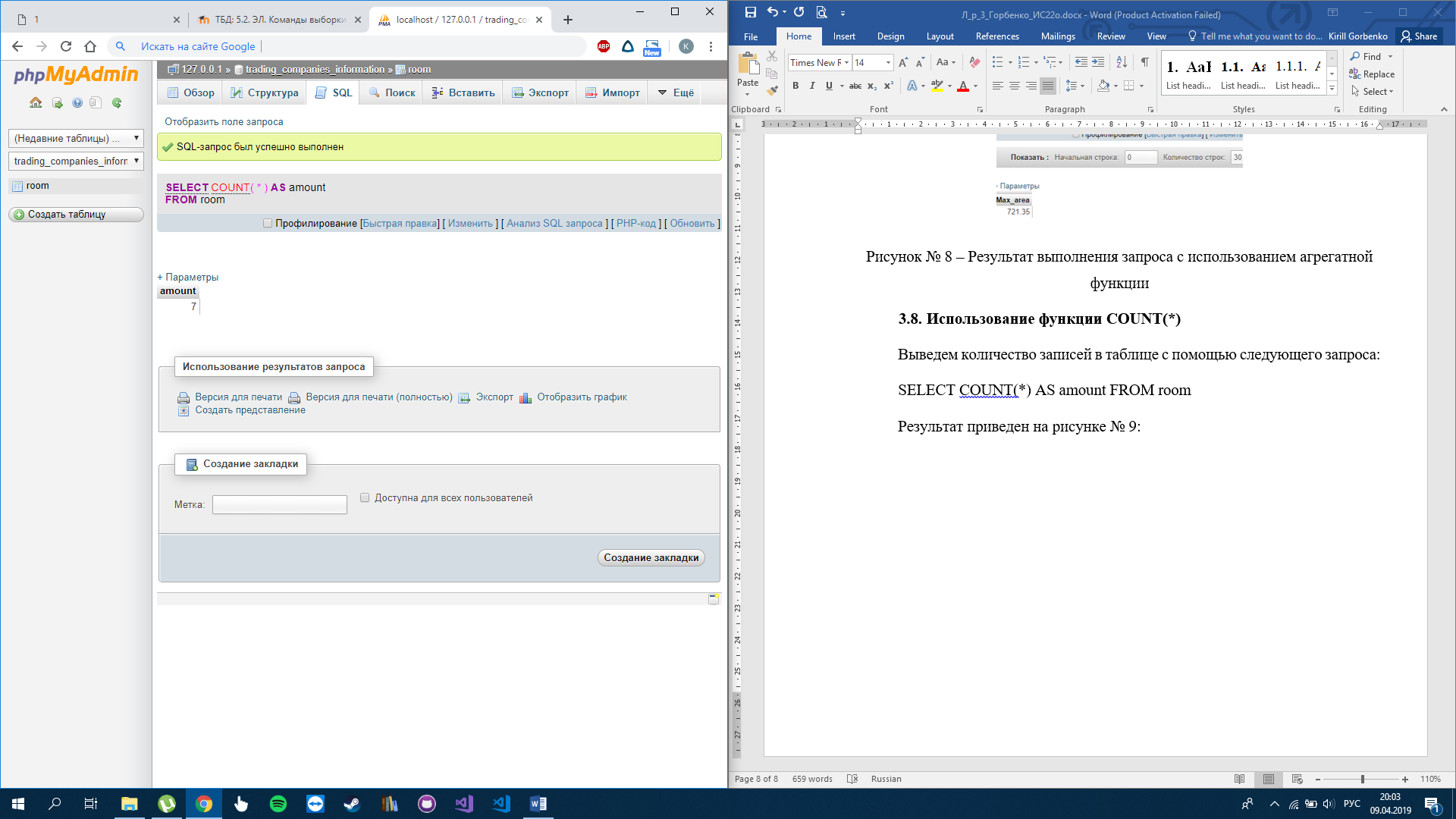


Рисунок № 9 – Результат запроса с использованием функции COUNT(\*)

* + 1. Использование простых вычислений в запросе

Выведем все данные из таблицы помещений, при этом значения поля «Тип комнаты» умножая на два:

SELECT Address, Number, Area, Room\_type \* 2 FROM room

Результат изображен на рисунке № 10:

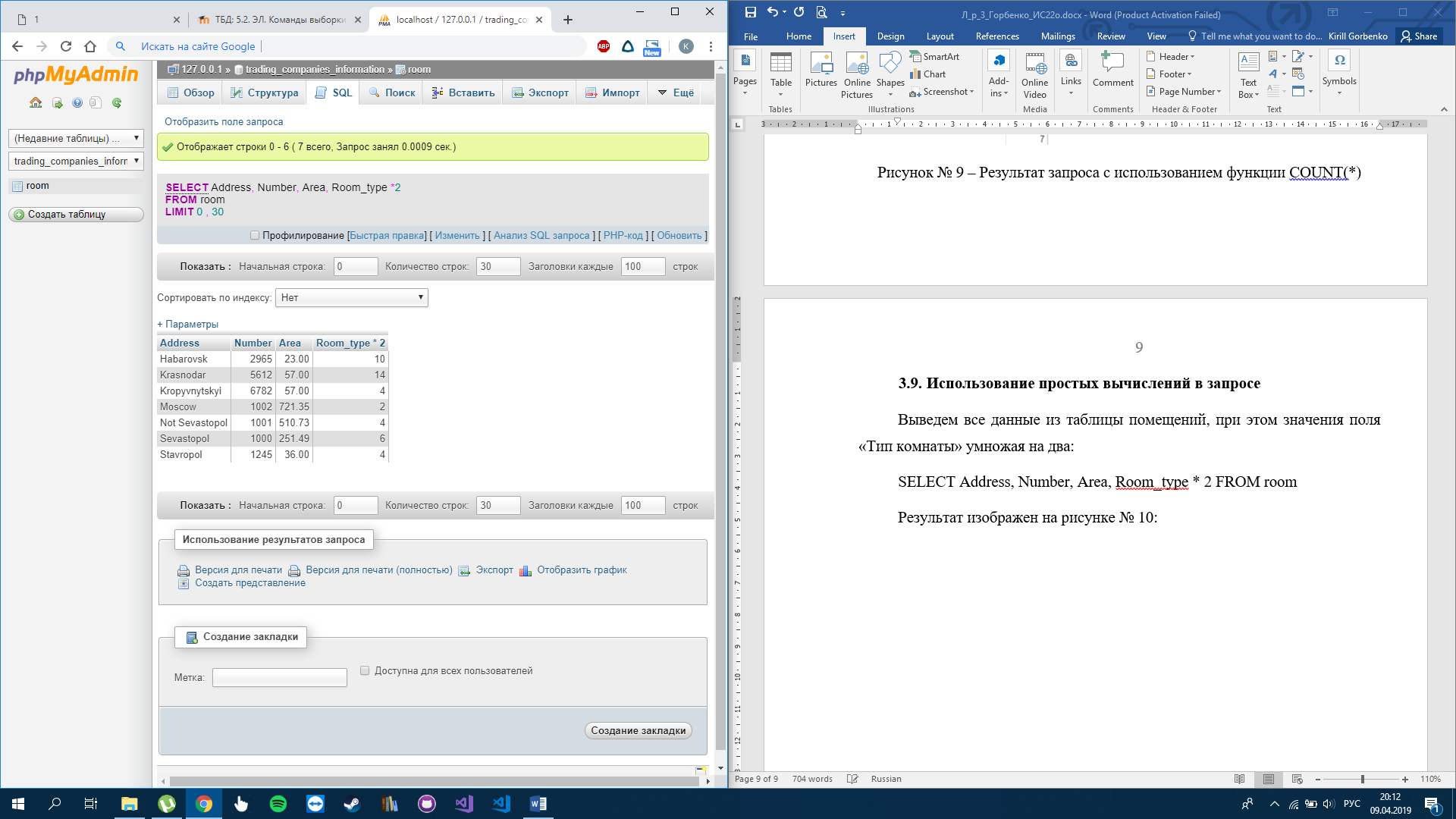


Рисунок № 10 – Результат запроса с использованием простых вычислений

* + 1. Использование простых вычислений как параметров функций

Выведем максимальное значение из квадратов типов комнаты:

SELECT MAX(Room\_type \* Room\_type) FROM room

Результат изображен на рисунке № 11:

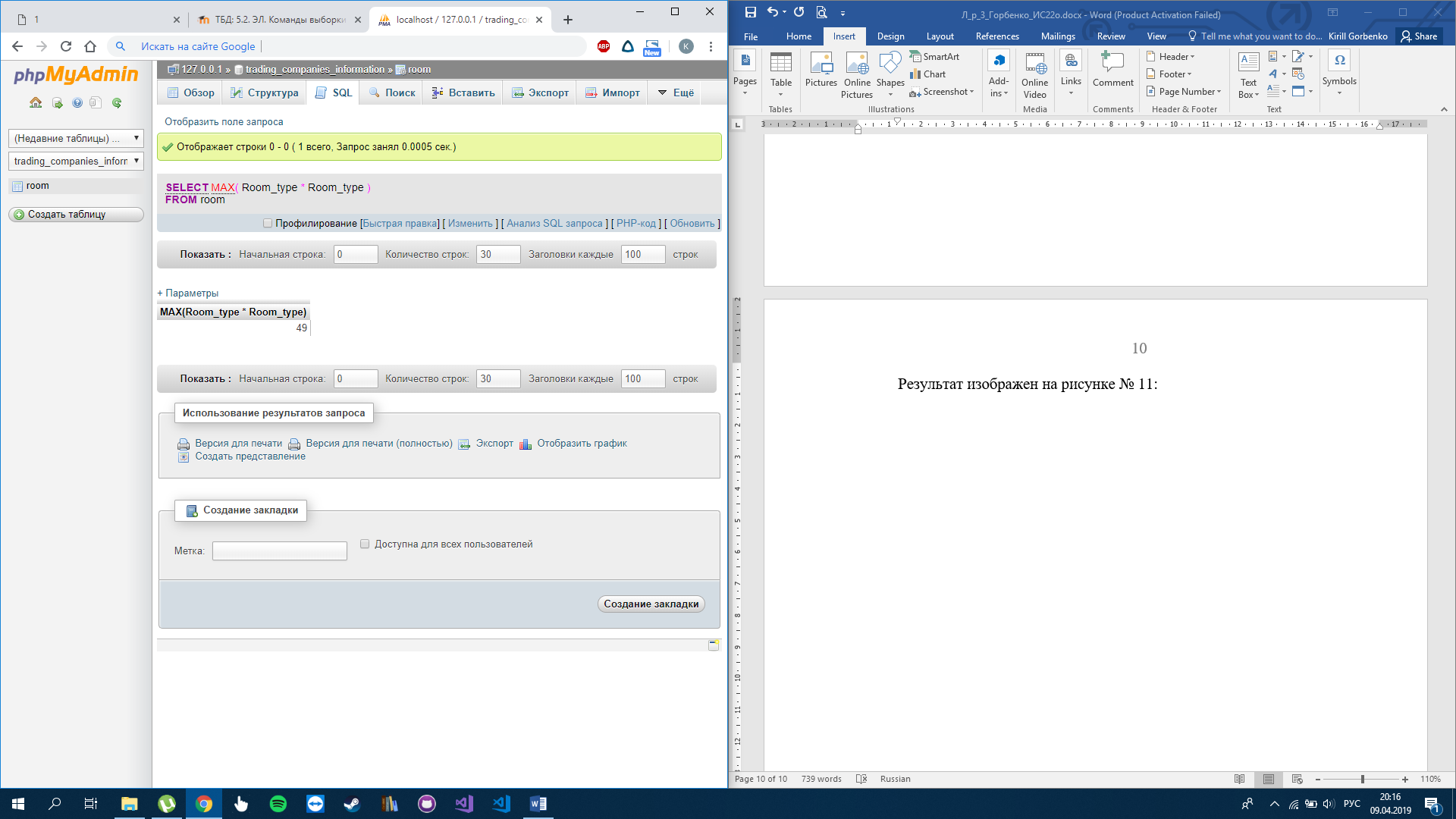


Рисунок № 11 – Результат выполнения запроса с использованием простых вычислений в качестве аргументов агрегатных функций

* + 1. Использование оператора GROUP BY

Выведем количества помещений всех каждого типа:

SELECT Room\_type, COUNT(\*) FROM room GROUP BY Room\_type

Результат представлен на рисунке № 12:

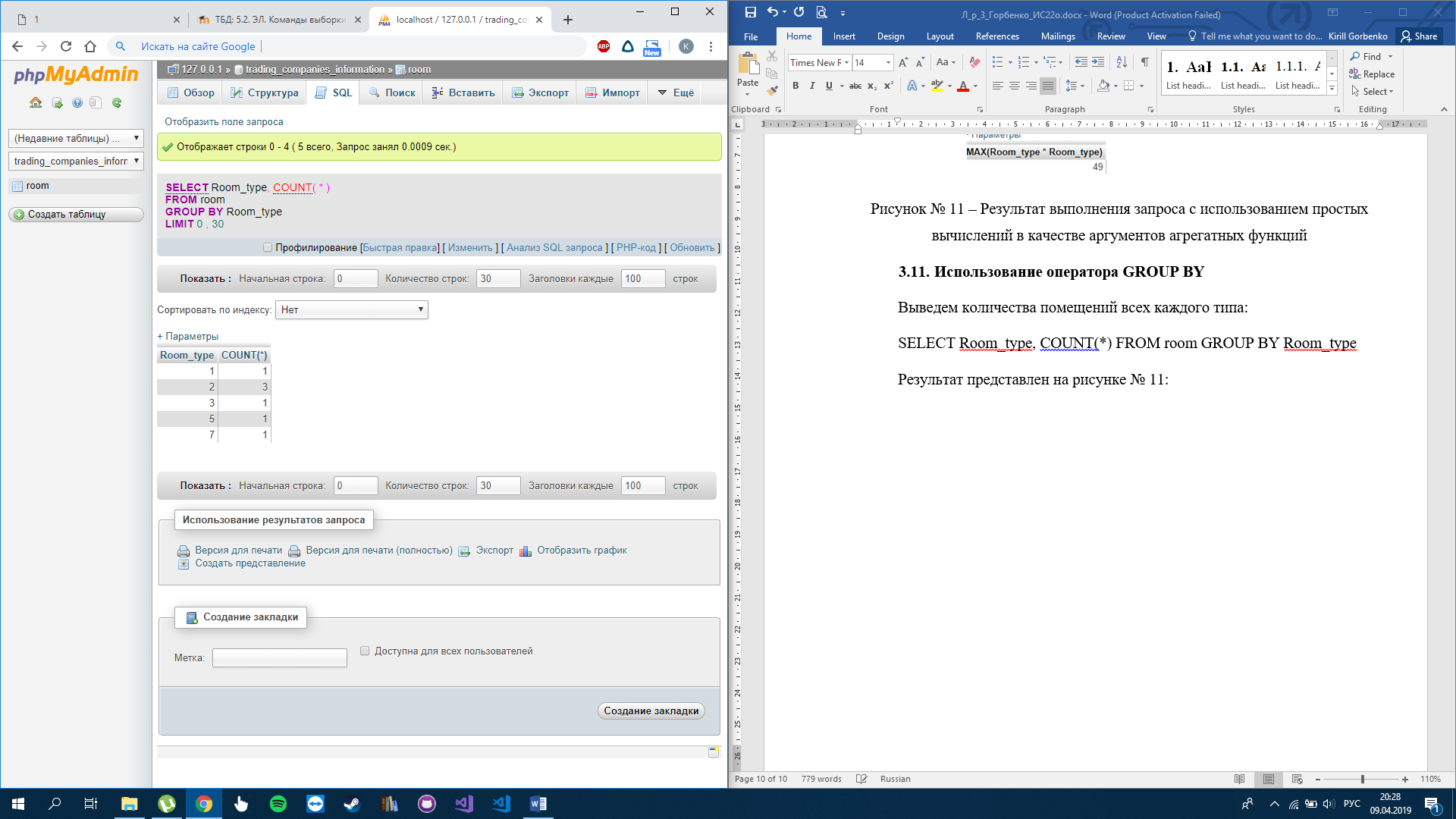


Рисунок № 12 – Результат выполнения запроса с использованием оператора GROUP BY

* + 1. Использование оператора HAVING

Выбрать сумму площадей и типы комнат, сумма площадей для всех комнат соответствующего типа больше 200:

SELECT Sum(Area),Room\_type FROM room GROUP BY Room\_type HAVING Sum(Area) > 200

Результат выполнения запроса представлен на рисунке № 13:

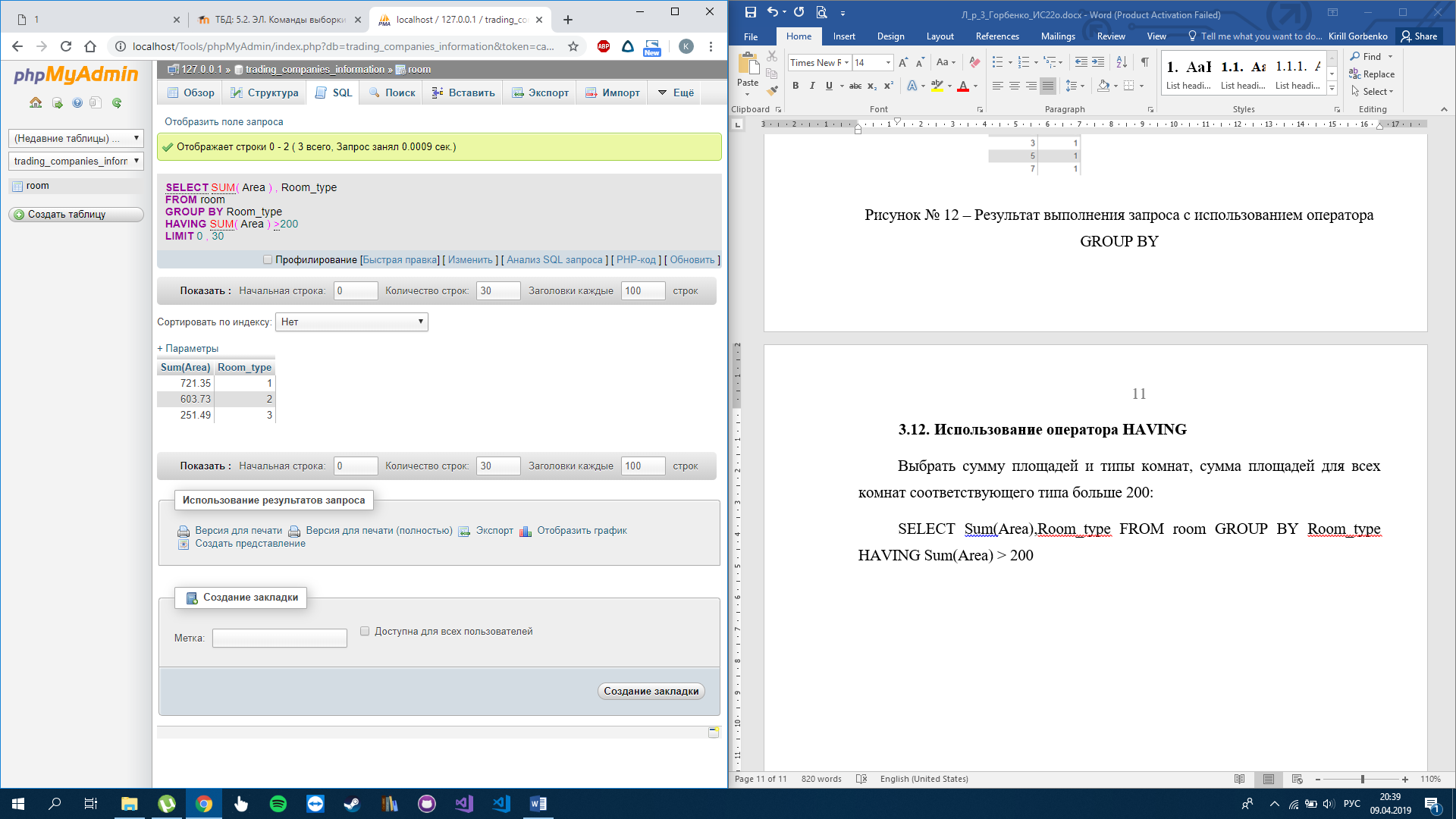


Рисунок № 13 – Результат выполнения запроса с использованием оператора HAVING

* + 1. Фильтрация информация

Осуществим фильтрацию данных в таблице по полю «Адрес». Выведем Всю информацию о помещениях, которые находятся в Севастополе:

SELECT \* FROM room WHERE Address = 'Sevastopol'

Результат выполнения запроса изображен на рисунке № 14:

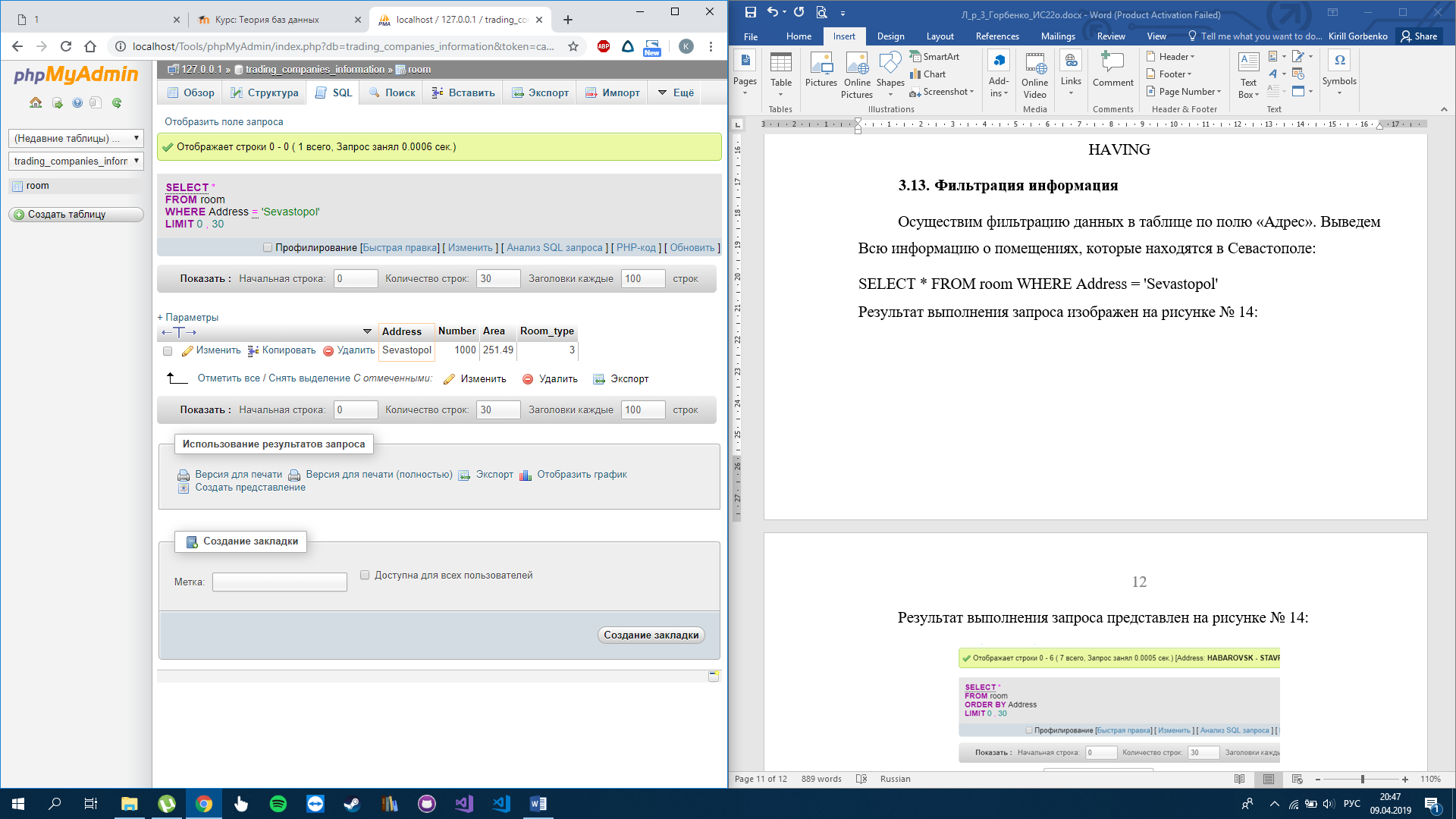


Рисунок № 14 – Результат запроса фильтрации по полю

* + 1. Применение аггрегатной функции к группам

Сгруппируем данные по полю «Номер фирмы» и применим функцию «Максимальная площадь». Выведем номера фирм и максимальные площади фирм с соответствующими номерами:

SELECT Number, Max(Area) FROM room GROUP BY Number

Результат выполнения запроса представлен на рисунке № 15:

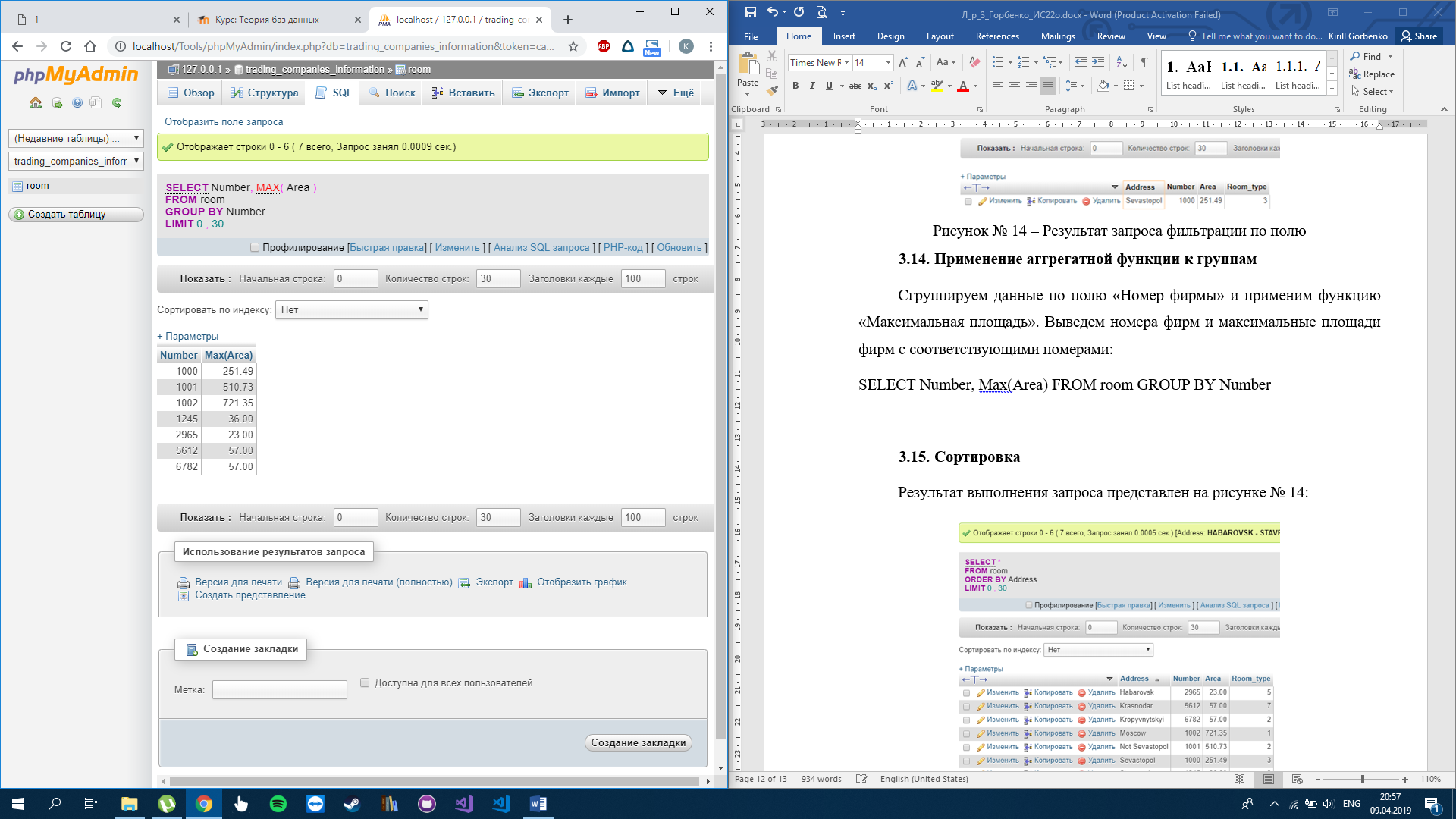


Рисунок № 15- Результат выполнения запроса

Такой запрос при имеющихся данных не имеет смысла так как каждая группа содержит по одному элементу.

* + 1. Сортировка

Выполним сортировку по убыванию по полю «Номер фирмы»:

SELECT \* FROM room ORDER BY Number DESC;

Результат выполнения запроса представлен на рисунке № 16:

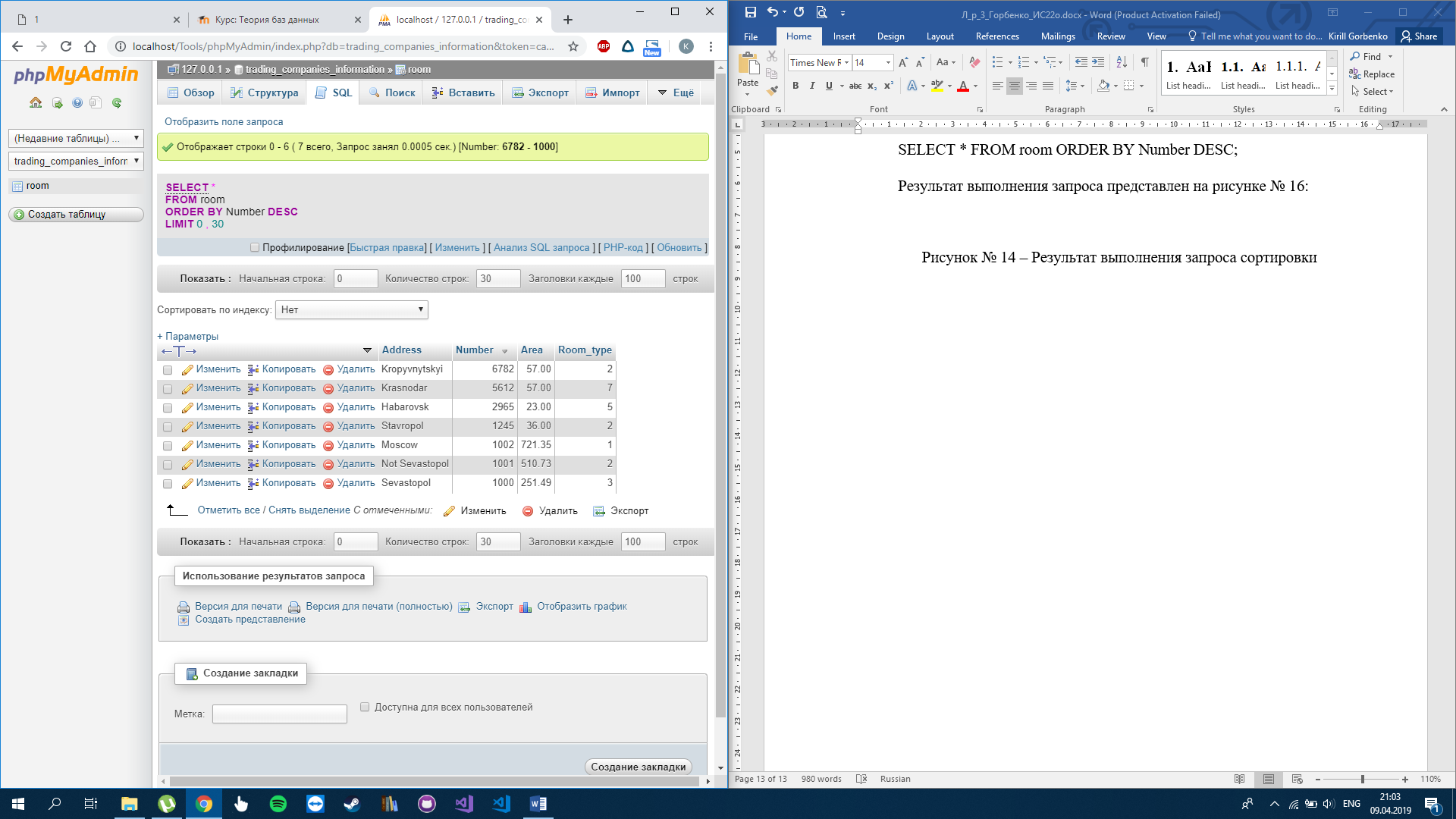


Рисунок № 16 – Результат выполнения запроса сортировки

* 1. ВЫВОД

В ходе лабораторной работы был изучен оператор выборки данных SELECT. С его помощью можно совершать выборку данных, простые вычисления, упорядочивание выводимых записей, группировка записей и применение к группам групповых операций.

Для вывода уникальных записей используется предикат DISTINCT.

Не имеет смысла осуществлять группировку по ключевым полям таблицы, так как такие группы будут иметь по одному элементу.