ОТЧЕТ

по лабораторной работе № 6

«Создание пользовательских представлений и индексов»

по дисциплине

«ТЕОРИЯ БАЗ ДАННЫХ»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверила:

Лебедева М.А.

* 1. ЦЕЛЬ РАБОТЫ

Ознакомиться с принципами работы пользовательских представлений и индексов, продемонстрировать работу с ними на примере.

* 1. ЗАДАНИЕ НА РАБОТУ
     1. Создание представления

Для заданной вариантом базы данных разработать и выполнить с помощью SQL-редактора PhpMyAdmin следующие запросы:

1. SQL-запрос на создание представления для ввода данных в таблицу (с использованием ограничений) ;
2. SQL-запрос, который продемонстрирует успешный ввод данных через созданное представление (с учетом заданных ограничений на ввод);
3. SQL-запрос, который продемонстрирует отрицательный результат при вводе данных через созданное представление (с учетом заданных ограничений на ввод) и объяснить причину;
4. SQL-запрос на удаление созданного представления;
5. SQL-запрос на создание представления для вывода неполных данных из двух или более таблиц одновременно.
   * 1. Создание индекса

Разработать и выполнить с помощью SQL-редактора PhpMyAdmin следующие запросы:

SQL-запрос на создание уникального индекса;

SQL-запрос на создание индекса на поле, по которому сортируются данные;

SQL-запрос на создание индекса на поле, по которому происходит группировка во время агрегации;

SQL-запрос на создание составного индекса с учетом селективности колонок (обосновать свой выбор);

SQL-запрос на создание составного индекса, причем обосновать очередность перечисления полей в индексе (группировка, сортировка, частота использования) и их количество.

После каждого запроса на создание индекса представить соответствующий запрос на выборку, для которого индексация имеет смысл. После чего обязательно удалять созданный индекс.

Заданная вариантом № 15 предметная область:

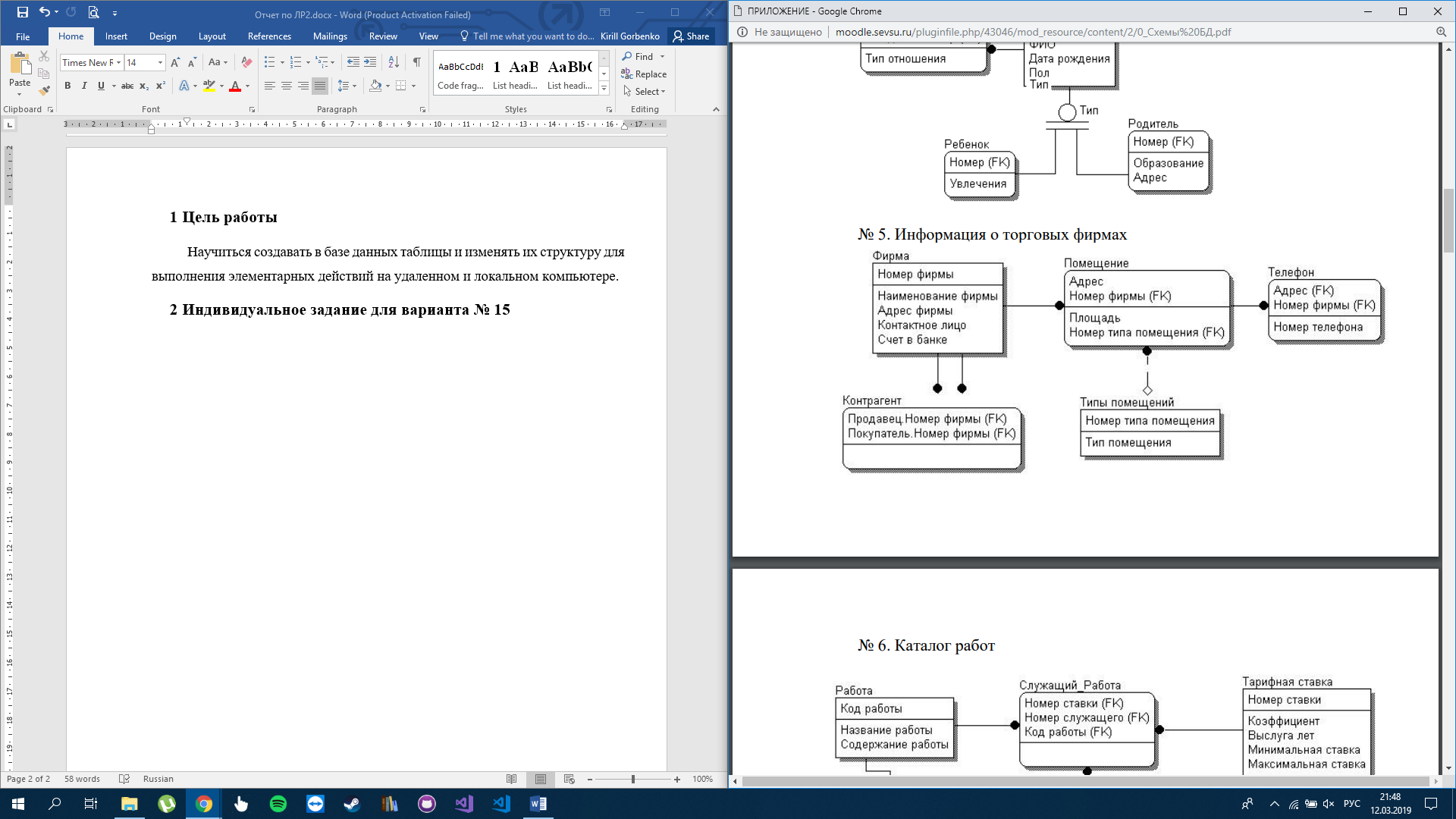


Рисунок 1 – Схема № 5 «Информация о торговых фирмах»

* 1. ХОД РАБОТЫ
     1. Создание представления для ввода данных

Создадим представление для ввода в таблицу «Фирма» данных «Номер фирмы», «Наименование фирмы» и «Адрес фирмы». При этом вводить в таблицу с помощью этого представления можно лишь записи, значение поля «Адрес» которых начинается с «Sevastopol».

CREATE VIEW company\_most\_important\_info

AS SELECT Number, Name, Address

FROM company

WHERE Address LIKE ('Sevastopol%')

WITH CHECK OPTION

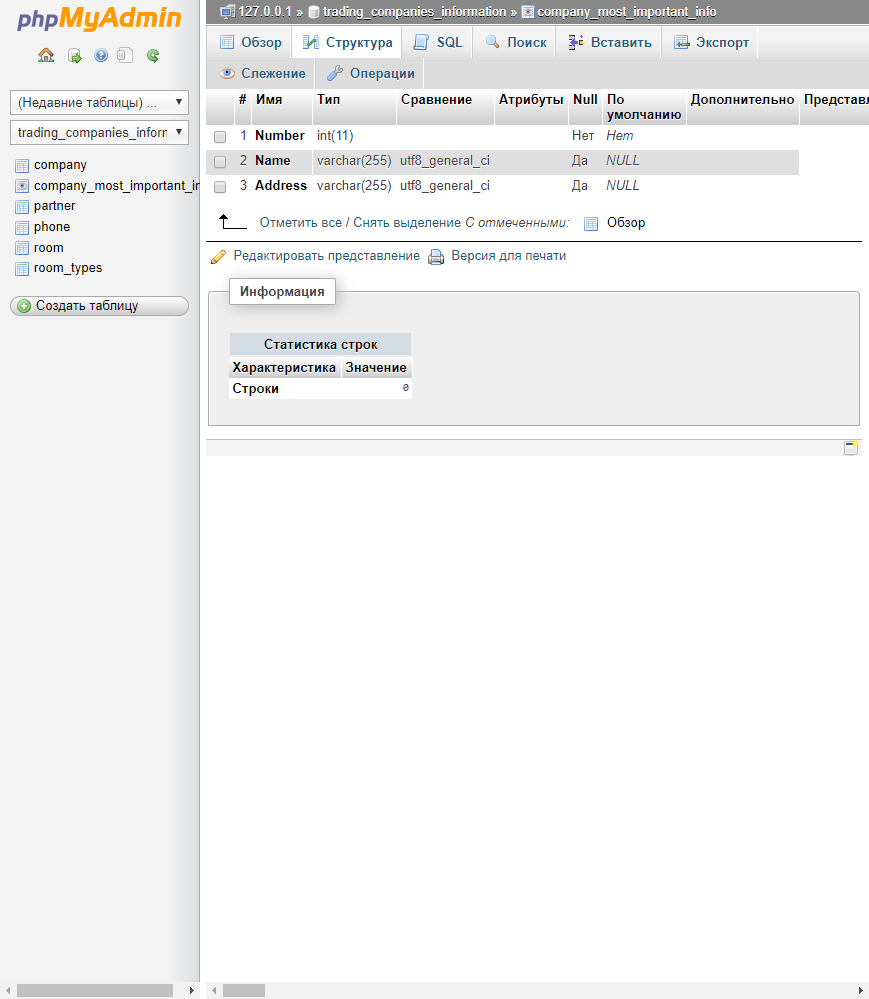


Рисунок 2 – Структура представления

* + 1. Запрос на ввод данных в таблицу посредством представления

Введем несколько записей в таблицу:

INSERT INTO company\_most\_important\_info VALUES

(2052, 'DNS', 'Sevastopol, Universitetskaya, 33'),

(5012, 'BurgerKing', 'Sevastopol, Lenina, 33')

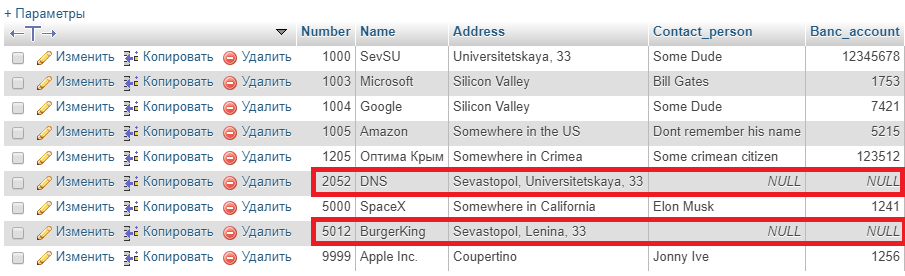


Рисунок 3 – Успешно введенные данные через представление

* + 1. Попытка ввести невалидные данные

Попробуем ввести в таблицу данные, значение поля «Адрес» которых не начинается на «Seastopol»:

INSERT INTO company\_most\_important\_info VALUES

(2102, 'McDonalds', 'NotSevastopol')

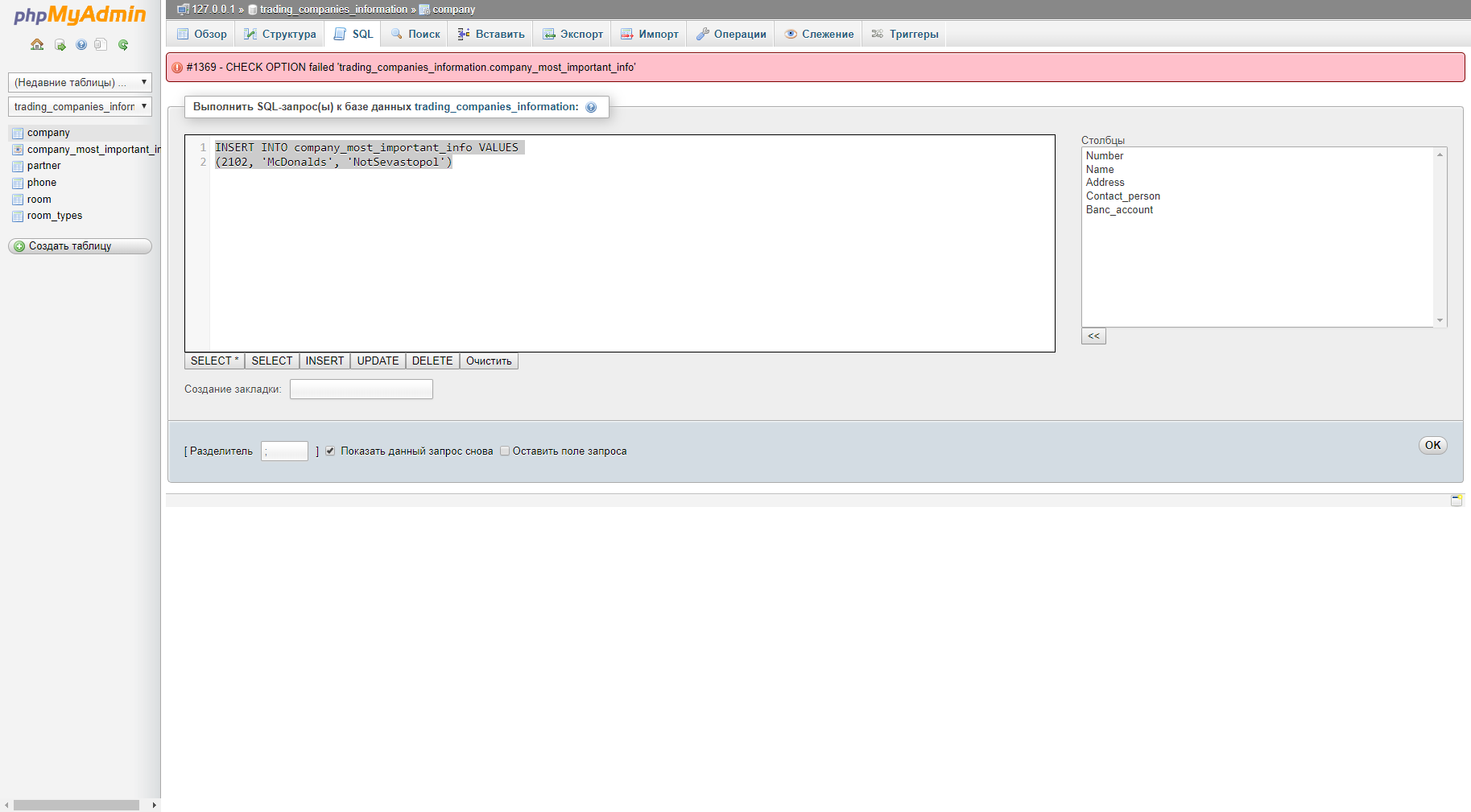


Рисунок 4 – Ошибка при попытке ввода неправильных данных

* + 1. Удаление представления

Для удаления представления используем следующий запрос:

DROP VIEW company\_most\_important\_info

* + 1. Создание представления для вывода данных

Создадим представление для вывода данных:

CREATE VIEW company\_room\_data

AS SELECT Name, company.Number, room.Address, Phone\_number

FROM (company JOIN room on company.Number = room.Number) JOIN phone ON room.Number = phone.Number AND room.Address = phone.Address

Данные, полученные с помощью представления:

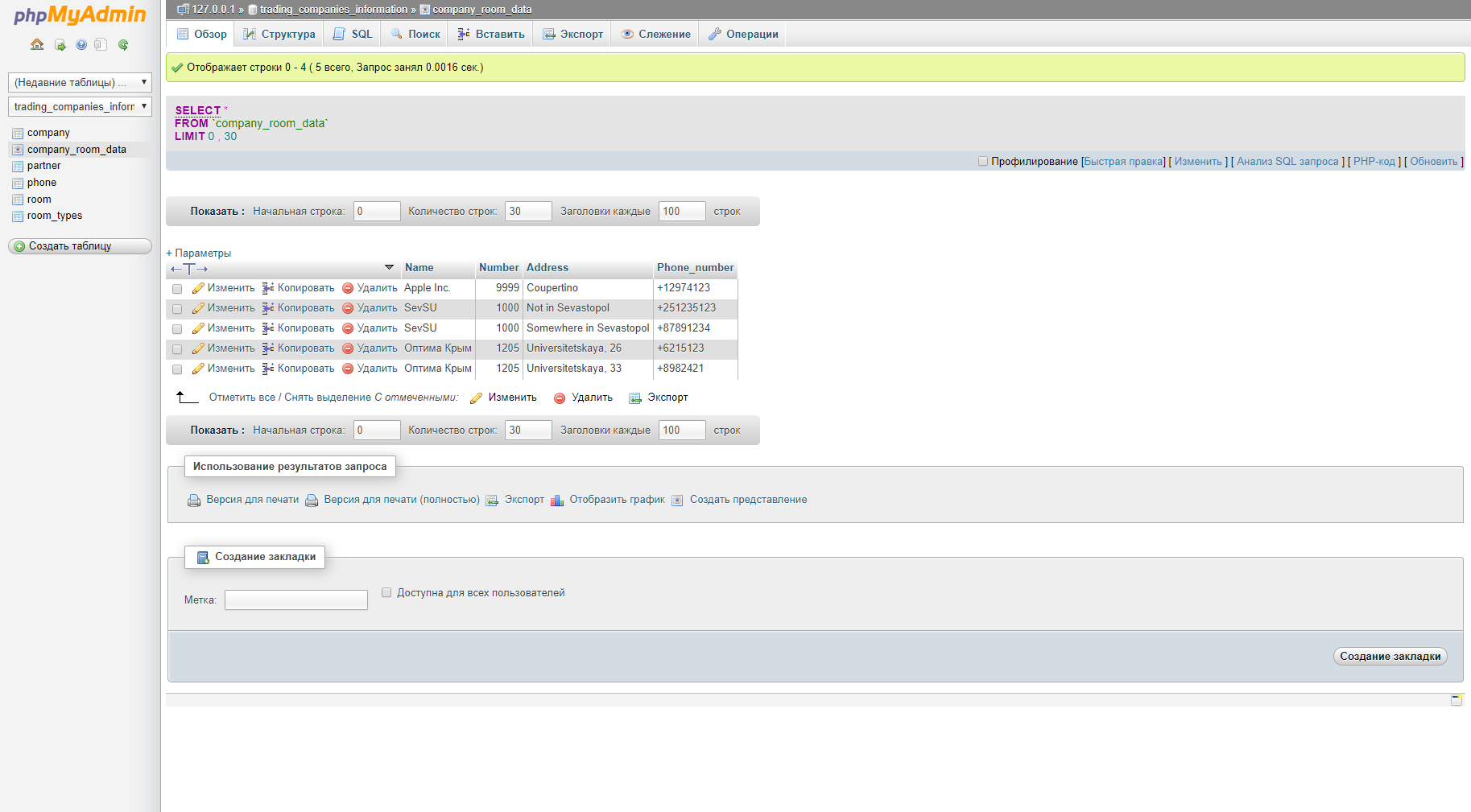


Рисунок 5 – Данные, полученные с помощью представления

* + 1. Создание уникального индекса

Создадим уникальный индекс:

CREATE UNIQUE INDEX unique\_company\_number ON room(Number)

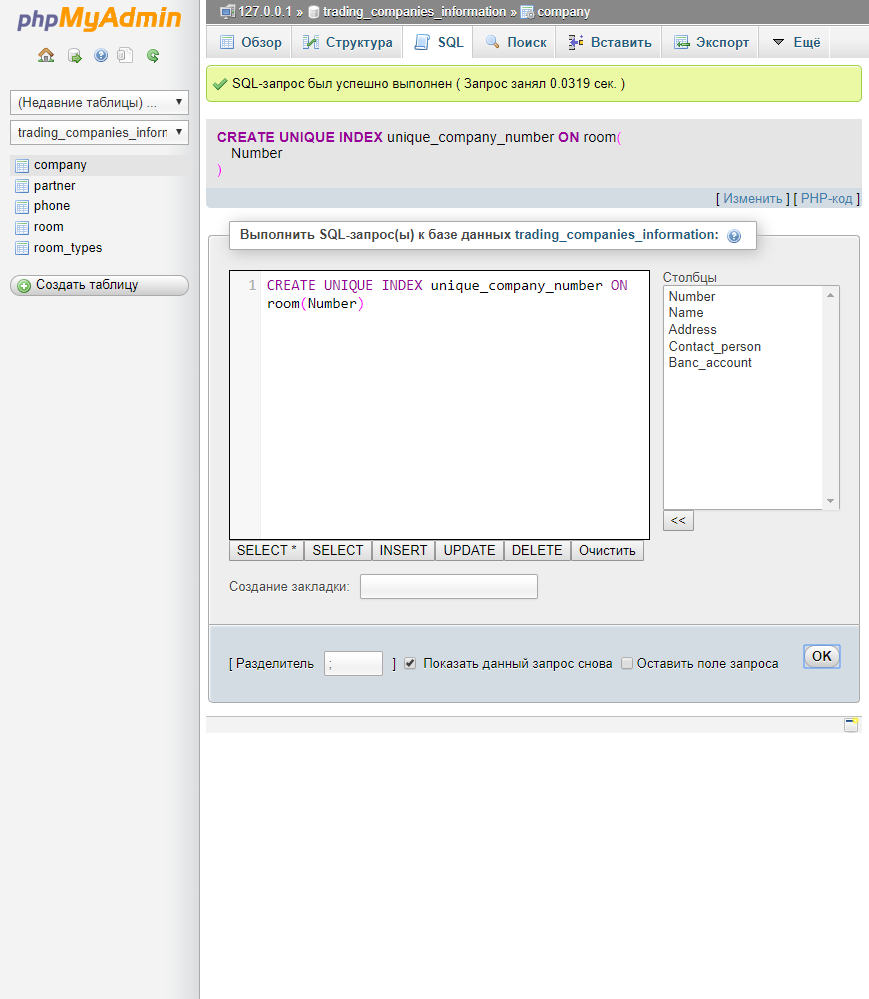


Рисунок 6 – Создание уникального индекса

Создадим запрос на выборку по индексу:

SELECT \* FROM room WHERE Number > 1000

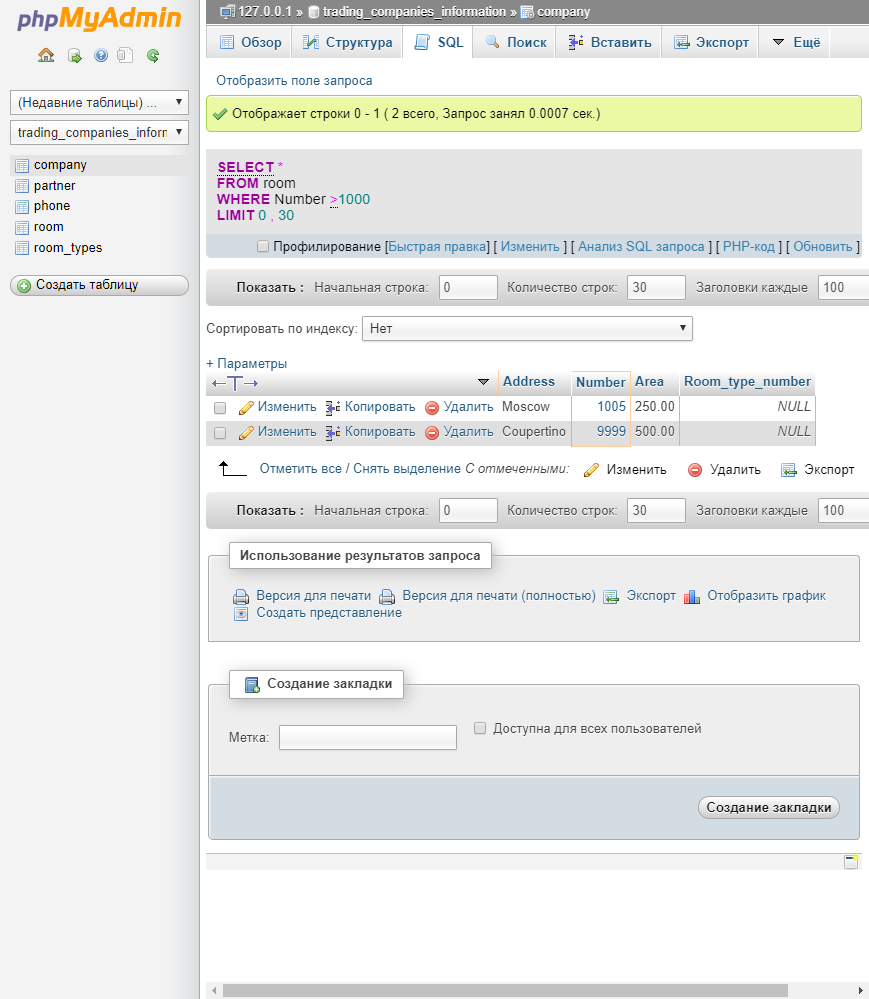


Рисунок 7 – Результат запроса на выборку по индексу

Удалим индекс:

DROP INDEX unique\_company\_number ON room

* + 1. Создание индекса на поле, по которому сортируются данные

Создадим индекс на сортировку помещений по типу комнаты:

CREATE INDEX room\_type\_number\_asc ON room(Room\_type\_number ASC)

SELECT \* FROM room ORDER BY Room\_type\_number ASC

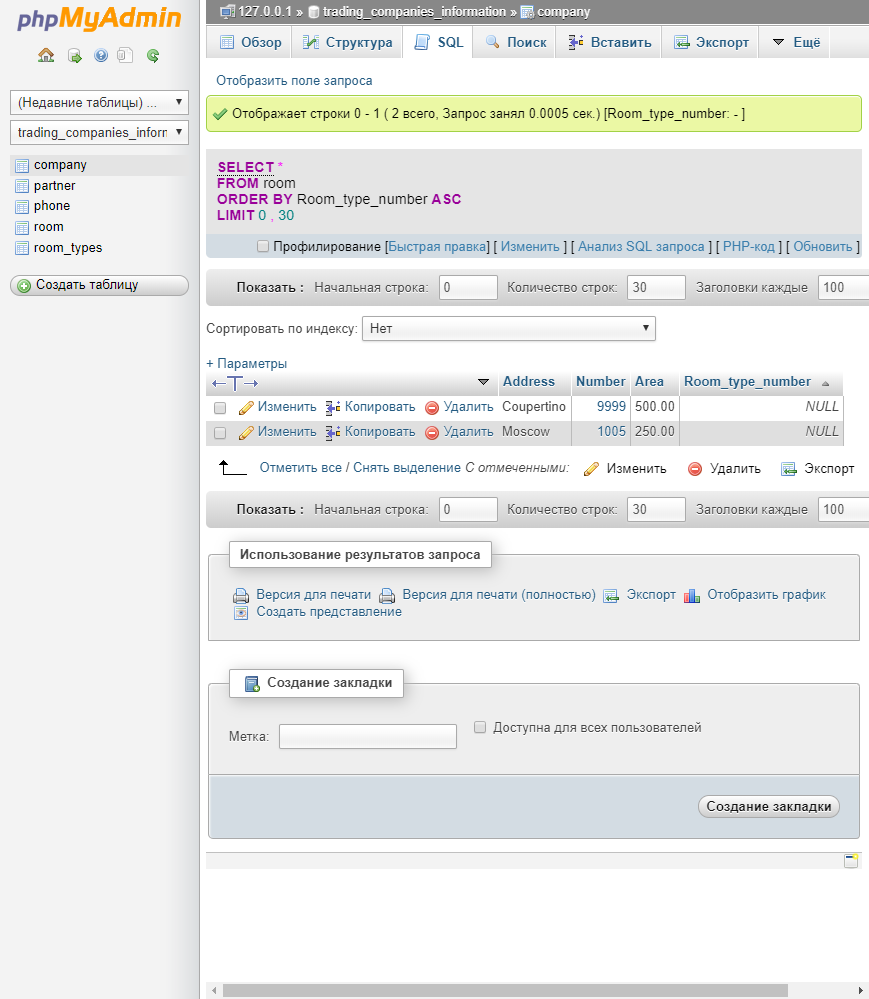


Рисунок 8 – Результат запроса на выборку по индексу

Удалим индекс:

DROP INDEX room\_type\_number\_asc ON room

* + 1. Создание индекса по полю, по которому происходит группировка

CREATE INDEX Room\_type\_number\_group ON room(Room\_type\_number)

Запрос на выборку типов помещений по индексу поля «Номер типа помещения».

SSELECT AVG(Area), Room\_type\_number FROM room

GROUP BY Room\_type\_number

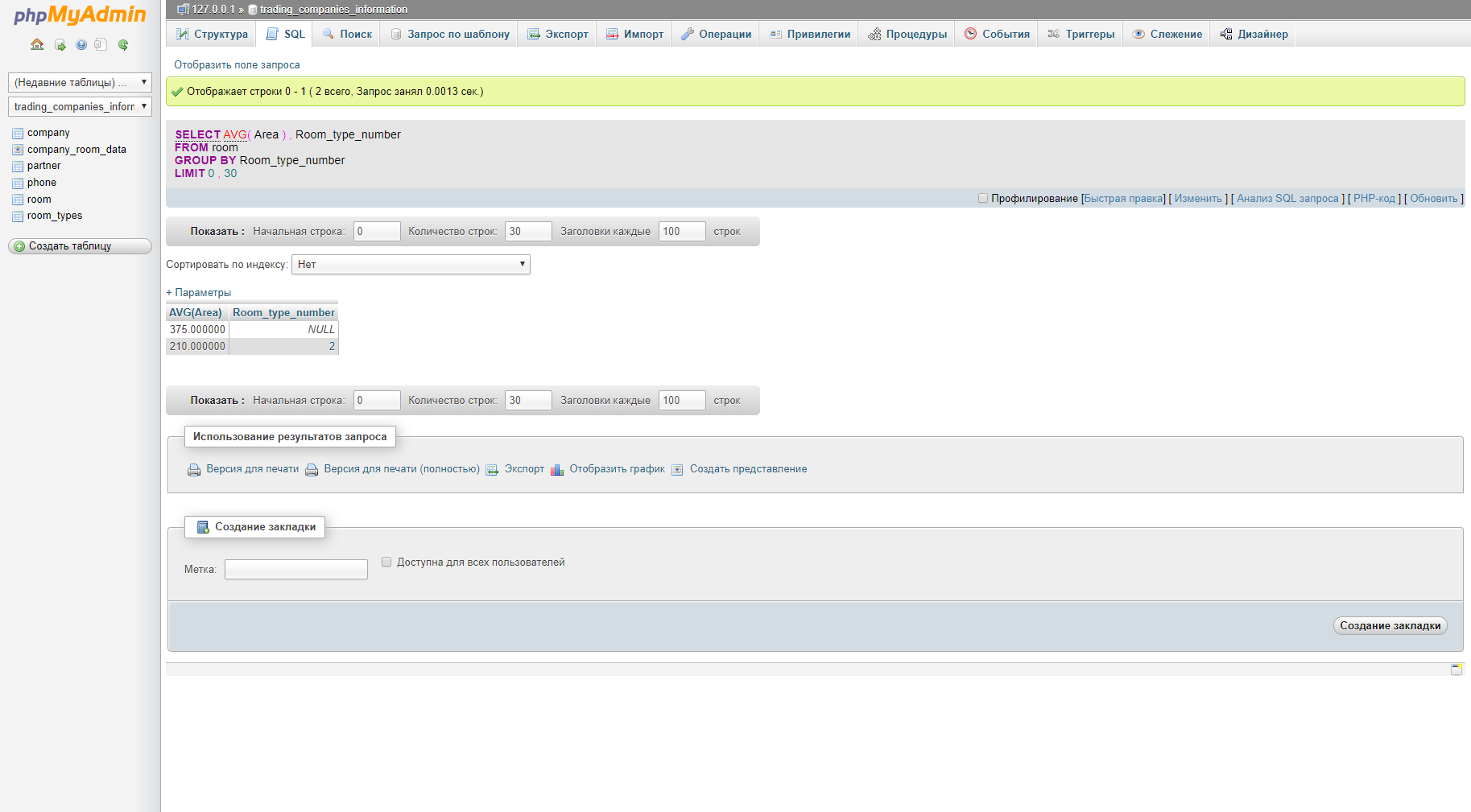


Рисунок 9 – Результат запроса на выборку

Удалим индекс:

DROP INDEX Room\_type\_number\_group ON room

* + 1. Создание составного индекса

CREATE INDEX roomType\_area ON room(Room\_type\_number ASC, Area ASC)

Запрос на выборку всех данных о помещениях, номер типа комнаты которых – 2 и площадь более 20. Для этого запроса имеет смысл создать сложный индекс c сортировкой полей по возрастанию, так как параметры условий имеют малые значения.

SELECT \* FROM room

WHERE Room\_type\_number = 2 AND Area > 20

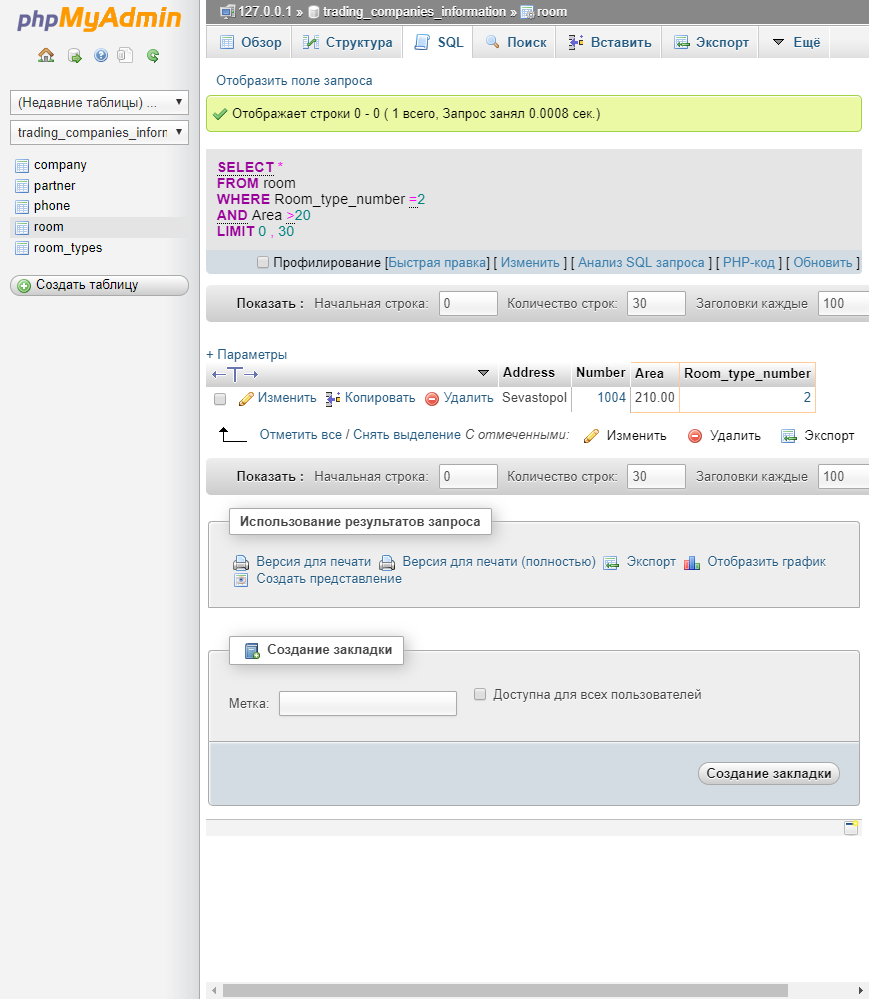


Рисунок 10 – Результат запроса на выборку по сложному индексу

Удалим индекс:

DROP INDEX roomType\_area ON room

* 1. ВЫВОД

В ходе лабораторной работы были изучены способы создания и использования представлений и индексов. Представления являются удобным способом многократного вызова часто используемого запроса, при этом не используя дисковое пространство.

При создании индекса необходимо продумывать, для каких запросов индекс будет использоваться и, исходя из этого, выбирать очередность и упорядоченность полей.