ОТЧЕТ

по лабораторной работе № 4

«Ссылочная целостность данных. Создание схемы базы данных»

по дисциплине

«ТЕОРИЯ БАЗ ДАННЫХ»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверила:

Лебедева М.А.

* 1. ЦЕЛЬ РАБОТЫ

Ознакомиться с принципами построения схемы базы данных и обеспечение целостности данных в базе.

* 1. ЗАДАНИЕ НА РАБОТУ

Проанализировать схему БД своего варианта задания (вариант тот же, что и в лабораторной работе № 1, 2, 3).

Выделить, классифицировать и описать все существующие связи, определить необходимые ограничения целостности.

С помощью SQL-запроса изменить существующую таблицу таким образом, чтобы она могла участвовать в связях с другими таблицами (например, с помощью оператора ALTER ТАВLE).

С помощью SQL-запросов создать еще не созданные таблицы. В процессе создания таблиц установить связи между таблицами.

Обязательно добавить во все таблицы записи (можно с помощью интерфейса phpMyAdmin).

Составить SQL-запросы на ввод данных в главную и подчиненную таблицу (выбрать любую пару таблиц). Проверить работу ограничений на значения первичного ключа обеих таблиц и внешнего ключа подчиненной таблицы.

Составить SQL-запросы на обновление и удаление данных для проверки работы ограничений целостности связей между таблицами. Проверить работу ограничений целостности в случаях установки каскадирования и запрета удаления и обновления данных.

Заданная вариантом № 15 предметная область:

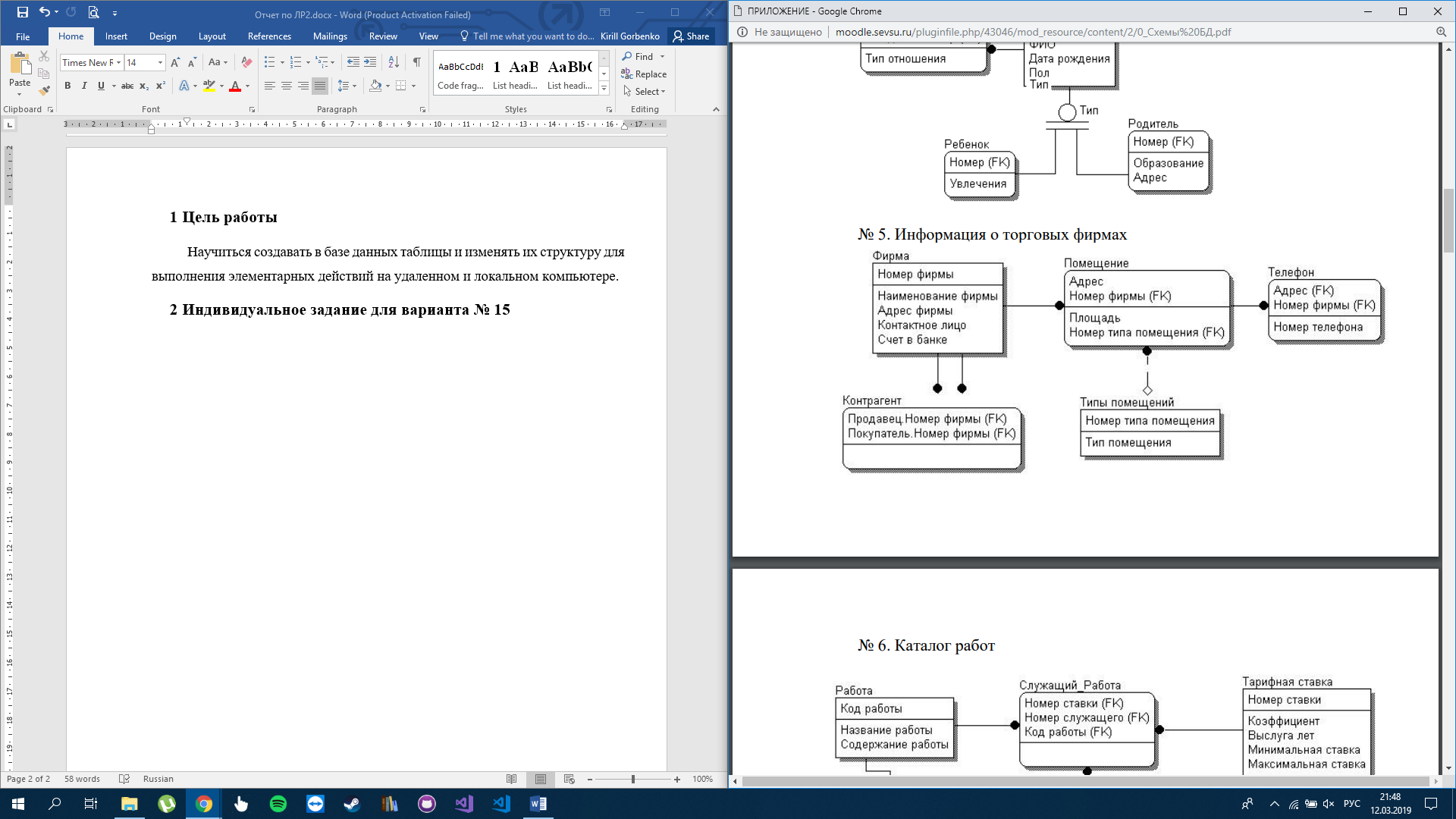


Рисунок № 1 – Схема № 5 «Информация о торговых фирмах»

* 1. ХОД РАБОТЫ
     1. Описание связей схемы

Начнем с таблицы «Помещение», уже имеющейся в базе данных. У этой таблицы сложный первичный ключ, состоящий из полей «Адрес» и «Номер фирмы». На этот первичный ключ ссылается внешний ключ таблицы «Телефон». Тип связи – один к одному, связь идентифицирующая. Главной таблицей является таблица «Помещение», дочерней – «Телефон».

Таблица «Типы помещений» имеет простой первичный ключ, включающий поле «Номер типа помещения». Он состоит в связи с внешним ключом таблицы «Помещение». Тип связи – 1 ко многим, связь неидентифицирующая. Главная таблица – «Типы помещений», дочерняя – «Помещение».

Таблица «Фирма» имеет первичный ключ, состоящий из поля «Номер фирмы». На этот первичный ключ ссылается внешний ключ таблицы «Помещение». Связь – один ко многим, связь идентифицирующая. Главная таблица – «Фирма», дочерняя – «Помещение».

Также, на первичный ключ таблицы «Фирма» ссылаются два внешних ключа таблицы «Контрагент»: поля «Продавец.Номер фирмы» и «Покупатель. Номер фирмы». Два этих поля составляют сложный первичный ключ таблицы «Контрагент». Связь между таблицами – один ко многим, идентифицирующая. Главной таблицей является таблица «Фирма», дочерней – «Контрагент».

* + 1. Создание таблицы «Телефон»

Создадим таблицу «Телефон» с помощью следующего запроса:

CREATE TABLE Phone (

Address varchar(80),

Number int NOT NULL,

Phone\_number varchar(10),

CONSTRAINT phone\_primary\_key PRIMARY KEY (Address, Number),

CONSTRAINT phone\_foreign\_key FOREIGN KEY (Address, Number) REFERENCES room(Address, Number) ON DELETE CASCADE ON UPDATE CASCADE

)

* + 1. Создание таблицы «Типы помещений»

Создадим таблицу «Типы помещений» с помощью следующего запроса:

CREATE TABLE room\_types (

Room\_type\_number TINYINT(3) PRIMARY KEY,

Room\_type VARCHAR(30)

)

Добавим внешний ключ к таблице «Помещение» так, чтобы он ссылался на первичный ключ таблицы «Типы помещений»:

ALTER TABLE room

ADD CONSTRAINT room\_type\_number\_foreign\_key

FOREIGN KEY (Room\_type\_number)

REFERENCES room\_types(Room\_type\_number)

ON DELETE SET NULL

ON UPDATE CASCADE

* + 1. Создание таблицы «Фирма»

Создадим таблицу с помощью следующего запроса:

CREATE TABLE Company (

Number INT PRIMARY KEY,

Name VARCHAR(255),

Address VARCHAR(255),

Contact\_person VARCHAR(255),

Banc\_account INT

)

Добавим ссылку на таблицу «Фирма» из таблицы «Помещение»:

ALTER TABLE room

ADD CONSTRAINT company\_number\_foreign\_key

FOREIGN KEY (Number)

REFERENCES company(Number)

ON UPDATE CASCADE ON DELETE RESTRICT

* + 1. Создание таблицы «Контрагент»

Создадим таблицу «Контрагент» и добавим внешний ключ, ссылающийся на таблицу «Фирма» с помощью следующего запроса:

CREATE TABLE partner (

Seller\_company\_number INT,

Buyer\_company\_number INT,

CONSTRAINT PRIMARY KEY (Seller\_company\_number, Buyer\_company\_number),

CONSTRAINT seller\_foreign\_key

FOREIGN KEY (Seller\_company\_number)

REFERENCES company(Number)

ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT Buyer\_foreign\_key

FOREIGN KEY (Buyer\_company\_number)

REFERENCES company(Number)

ON DELETE RESTRICT ON UPDATE CASCADE

)

* + 1. Заполнение таблиц

Заполним родительскую таблицу «Типы помещений»:

INSERT INTO room\_types (Room\_type\_number, Room\_type)

VALUES (1, 'Office'),

(2, 'Storage'),

(3, 'Public service'),

(4, 'Private sector')

Заполним таблицу «Фирма»:

INSERT INTO company VALUES

(1000, 'SevSU', 'Universitetskaya, 33', 'Some Dude', 12345678),

(1001, 'SpaceX', 'Somewhere in California', 'Elon Musk', 1241),

(1002, 'Apple Inc.', 'Coupertino', 'Jonny Ive', 1256),

(1003, 'Microsoft', 'Silicon Valley', 'Bill Gates', 1753),

(1004, 'Google', 'Silicon Valley', 'Some Dude', 7421),

(1005, 'Amazon', 'Somewhere in the US', 'Dont remember his name', 5215)

Заполним таблицу «Контрагент»:

INSERT INTO PARTNER VALUES

(1000, 1001),

(1003, 1005)

Заполним таблицу «Помещение»:

INSERT INTO room VALUES

('Somewhere in Sevastopol', 1000, 100500, 3),

('Not in Sevastopol', 1000, 63, 2),

('Coupertino', 1002, 500, 1),

('Moscow', 1005, 250, 1)

Заполним таблицу «Телефон»:

INSERT INTO phone VALUES

('Somewhere in Sevastopol', 1000, '+87891234'),

('Not in Sevastopol', 1000, '+251235123'),

('Coupertino', 1002, '+12974123')

* + 1. Проверка ограничений первичного и внешнего ключей

Для проверки выберем таблицы «Фирма» и «Помещение». Главной таблицей является «Фирма». Ее главным ключом является поле «Номер фирмы». Попробуем добавить в дочернюю таблицу сущность, которая ссылается на значение первичного ключа, которого нет в главной таблице:

INSERT INTO ROOM VALUES

('Coupertino', 1006, 500, 2)

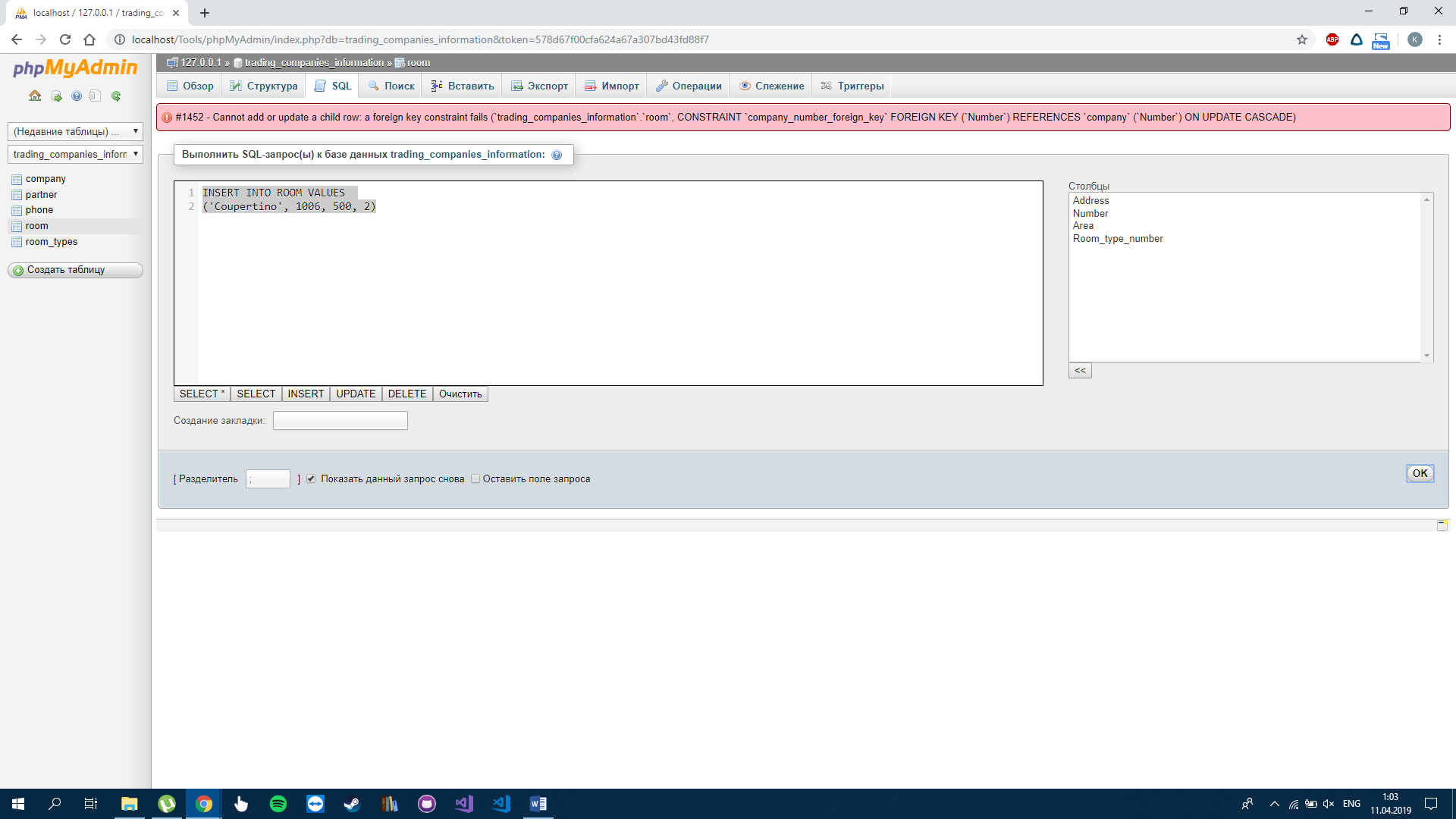


Рисунок № 2 – Ошибка при выполнении запроса, нарушающего целостность вторичного ключа

Попробуем добавить в таблицу «Фирма» запись с неуникальным значением ключевого поля:

INSERT INTO company VALUES

(1003, 'Some company', 'Some adress', 'Some person', 1234)

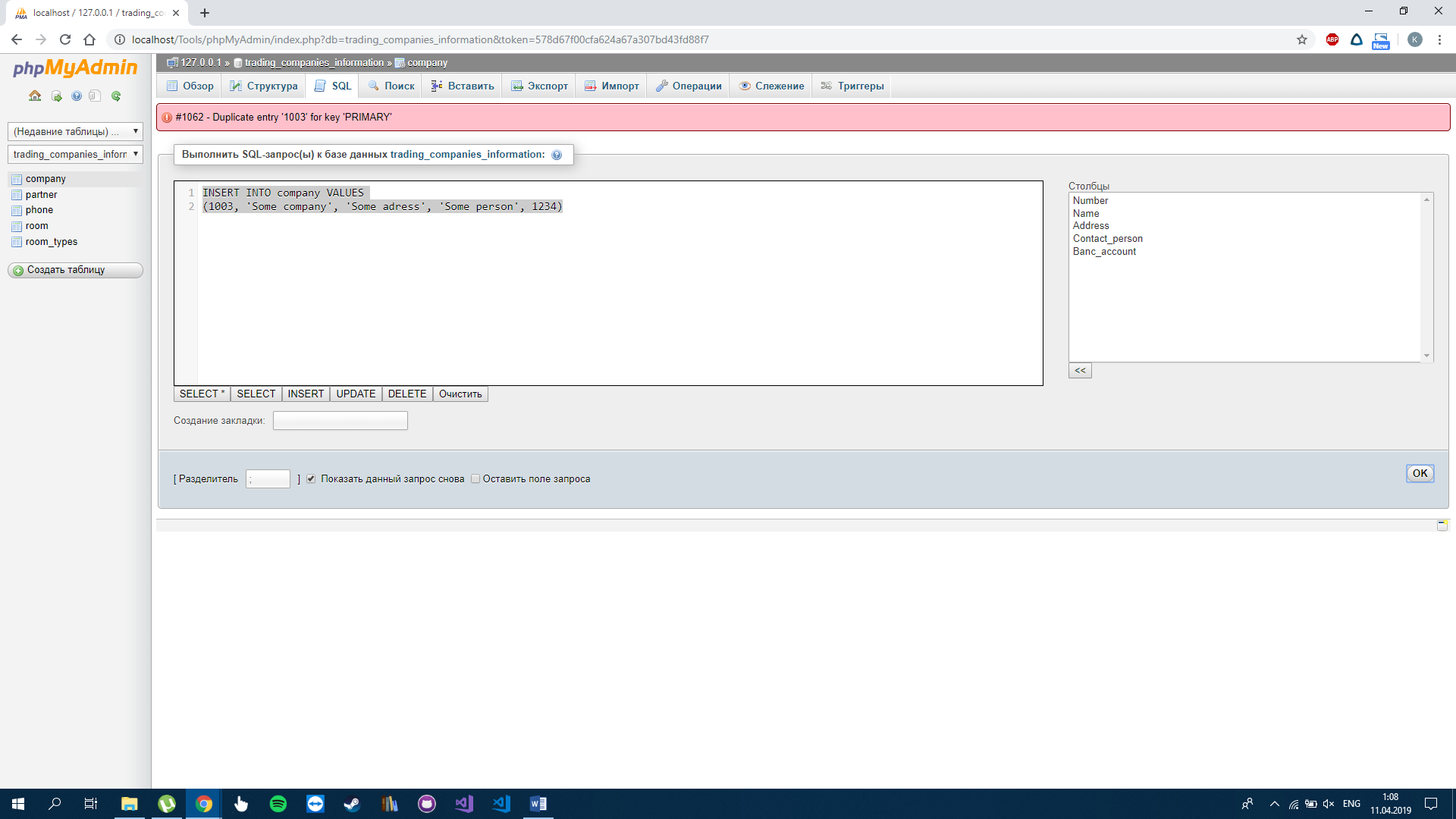


Рисунок № 3 – Ошибка при выполнении запроса, нарушающего целостность первичного ключа таблицы «Фирма»

Проверим целостность первичного ключа таблицы «Помещение»:

INSERT INTO room VALUES

('Somewhere in Sevastopol', 1000, 500, 2)

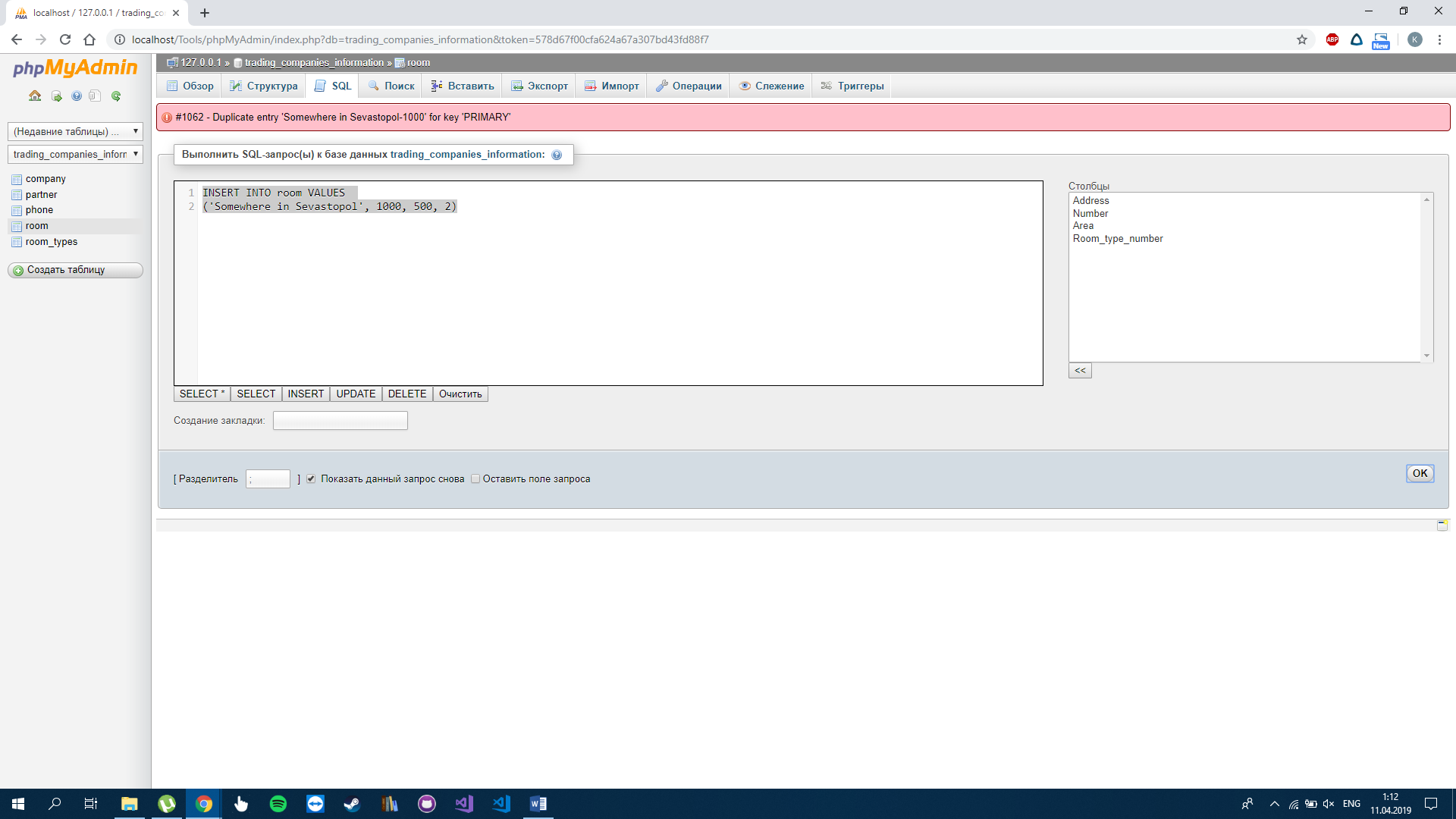


Рисунок № 4 – Ошибка при выполнении запроса, нарушающего целостность первичного ключа таблицы «Помещение»

* + 1. Проверка целостности при удалении и обновлении данных

При удалении из таблицы «Типы помещений» значения соответствующих полей таблицы «Помещение» должны установиться в NULL. Проверим:

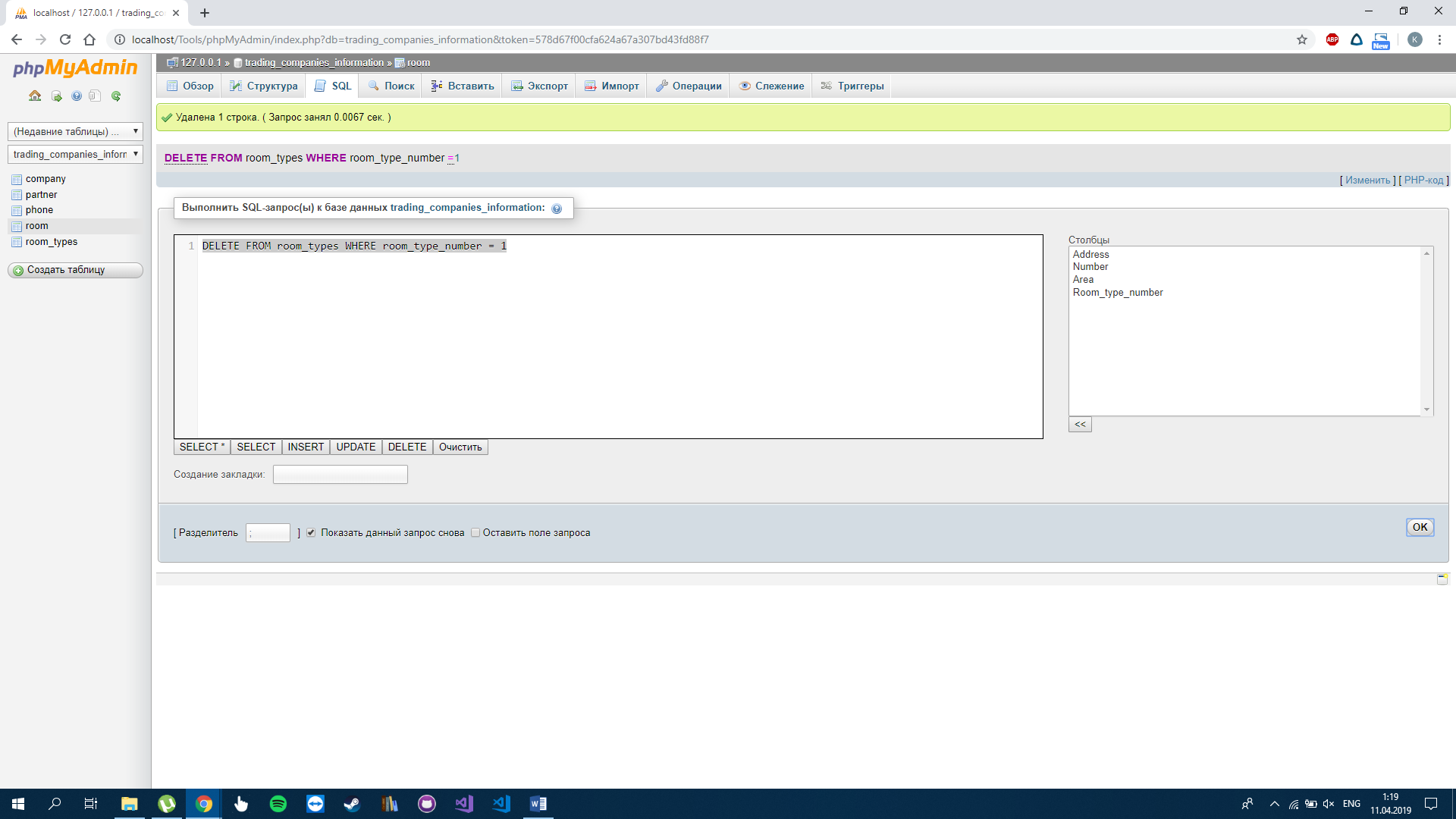


Рисунок № 5 – Успешный запрос на удаление из таблицы «Типы помещений»

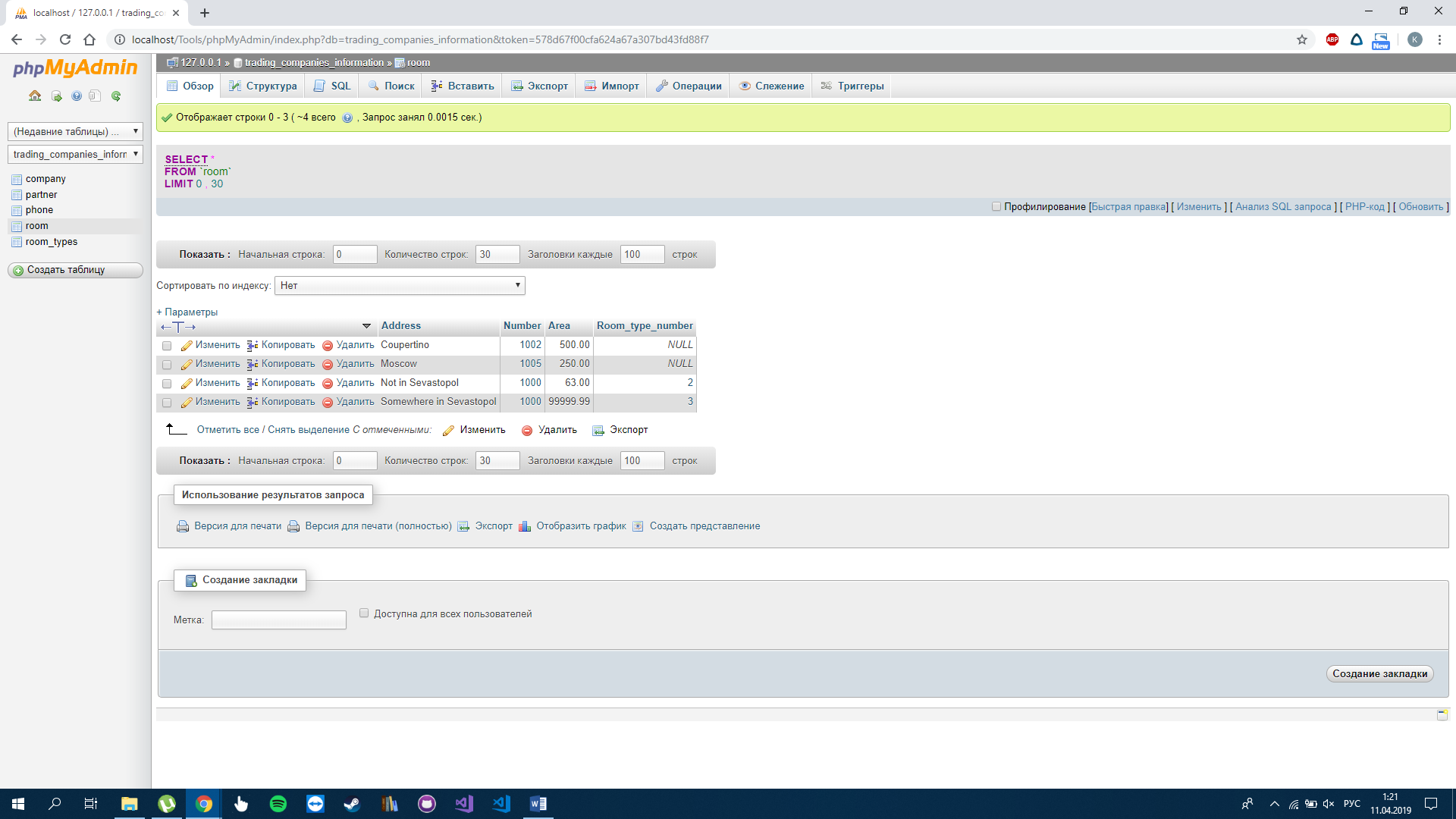


Рисунок № 6 – Состояние таблицы «Помещение» после запроса

Попробуем обновить ключевое поле главной таблицы «Фирма». В этом случае все ссылающиеся на него внешние ключи должны каскадно обновиться. Проверим:

UPDATE company SET Number = 9999 WHERE Number = 1002

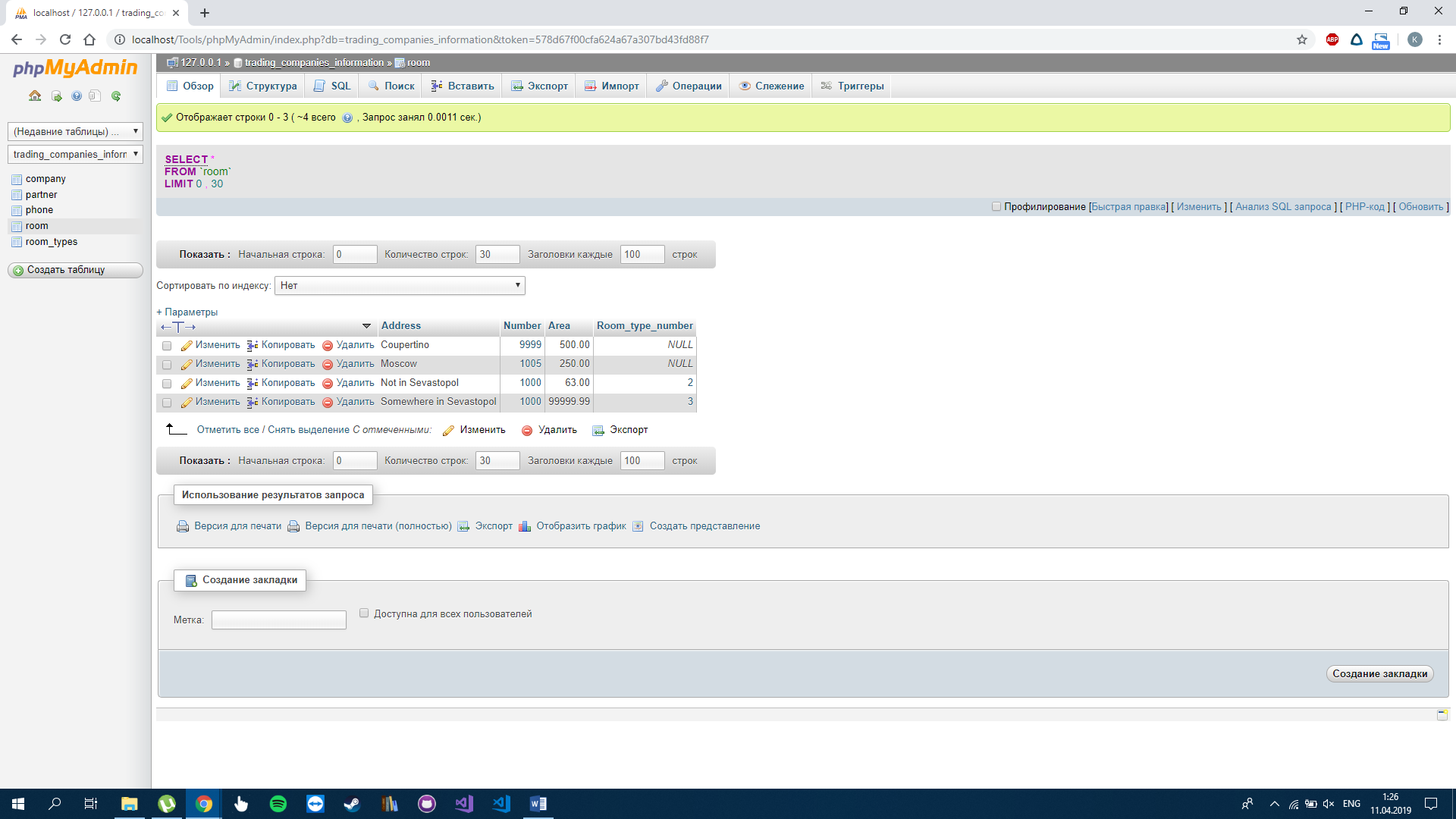


Рисунок № 7 – Состояние дочерней таблицы «Помещение» после обновления ключевого поля в главной таблице

Из рисунка № 7 видно, что в дочерней таблице «Помещение» значение поля номера изменилось.

При удалении записей из таблицы «Фирма» в случае, когда на них есть ссылки из таблицы «Контрагент», запрос должен блокироваться. Проверим это:

DELETE FROM company WHERE Number = 1000

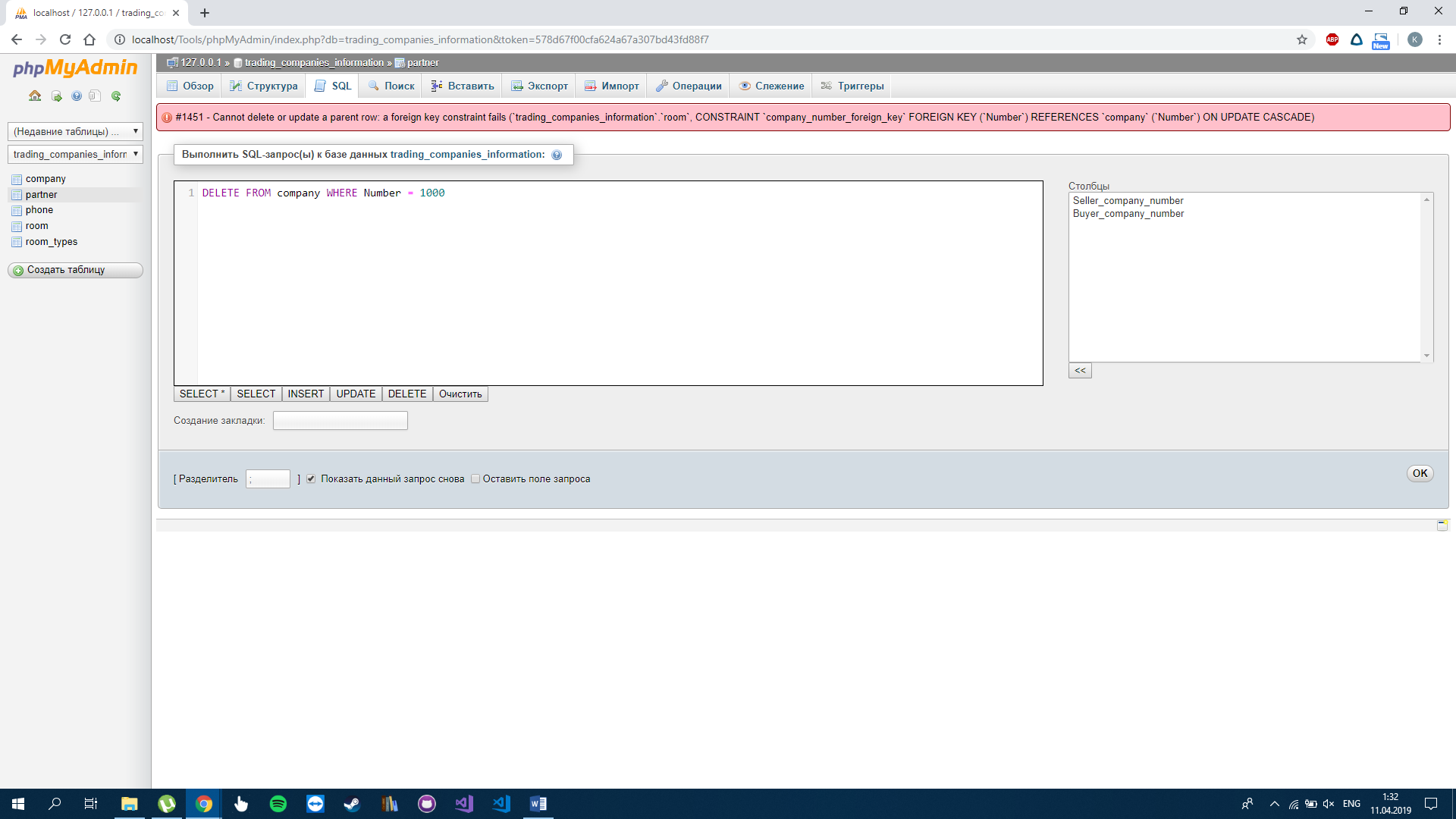


Рисунок № 8 – Ошибка при запросе

* 1. ВЫВОД

В ходе лабораторной работы была полностью реализована заданная база данных. Для этого были созданы недостающие после выполнения предыдущих трех лабораторных работ таблицы и связи между ними.

Для соблюдения целостности данных были обработаны ситуации, связанные с удалением или изменением данных, от которых зависят другие данные.