МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Сирота Марина Романовна

Институт информационных технологий и управления в технических системах

курс 3 группа ГИС/б-22-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №1

по дисциплине: «Управление данными»

по теме: «Манипулирование базой данных. Реляционная алгебра и SQL»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст. пр. Гончаренко Д.Г.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2018

1 ЦЕЛЬ РАБОТЫ

Изучить основы реляционной алгебры как базового средства манипулирования. Научиться представлять запросы как на реляционной алгебре, так и на SQL.

2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Проанализировать схему БД своего варианта задания (см. приложение А), выделить и классифицировать все существующие связи, определить необходимые ограничения целостности. Номер варианта (х) определяется как x = N mod 20, где N – номер студента в группе.

2. Создать базу данных и все ее таблицы. Особое внимание надо уделить описанию первичного ключа, значений по умолчанию, описателям NOT NULL и конструкции CHECK.

3. Установить связи между таблицами.

4. Занести в таблицу образцы данных оператором INSERT INTO. Необходимо занести не менее 10 строк.

5. Проверить работу ограничений целостности (каскадирование удаления, модификации и др.)

6. Продемонстрировать выполнение простых вычислений в запросе.

7. Использовать простое вычисление как параметр агрегатной функции.

8. Продемонстрировать работу предложения GROUP BY.

9. Продемонстрировать работу предложения HAVING.

10. Ознакомиться с операциями реляционной алгебры.

11. Применить к БД операции селекции и соединения в одном запросе.

12. Создать запрос, использующий операции проекции и деления (в одном запросе).

13. Создать запрос, использующий операции проекции, объединения и конъюнкции (в одном запросе).

14. Создать запрос, использующий операции соединения и деления (в одном запросе).

15. Создать запрос, использующий операции вычитания и дизъюнкции (в одном запросе).

16. Сформулировать и записать запрос на SQL, не реализующийся на РА.

Вариант 17



Рисунок 2.1 – Вариант

3 ВЫПОЛНЕНИЕ РАБОТЫ

1. Создание таблицы Клиент

CREATE TABLE `kp\_yd`.`Клиент` ( `Номер паспорта` INT(6) PRIMARY KEY , `ФИО` VARCHAR( 50 ), )

ENGINE = InnoDB;

1. Создание связи между таблицами Номер и Заказ:

ALTER TABLE `заказ` ADD FOREIGN KEY (`Порядковый номер`) REFERENCES `kp\_ud`.`номер`(`Порядковый номер`) ON DELETE CASCADEON UPDATE CASCADE;

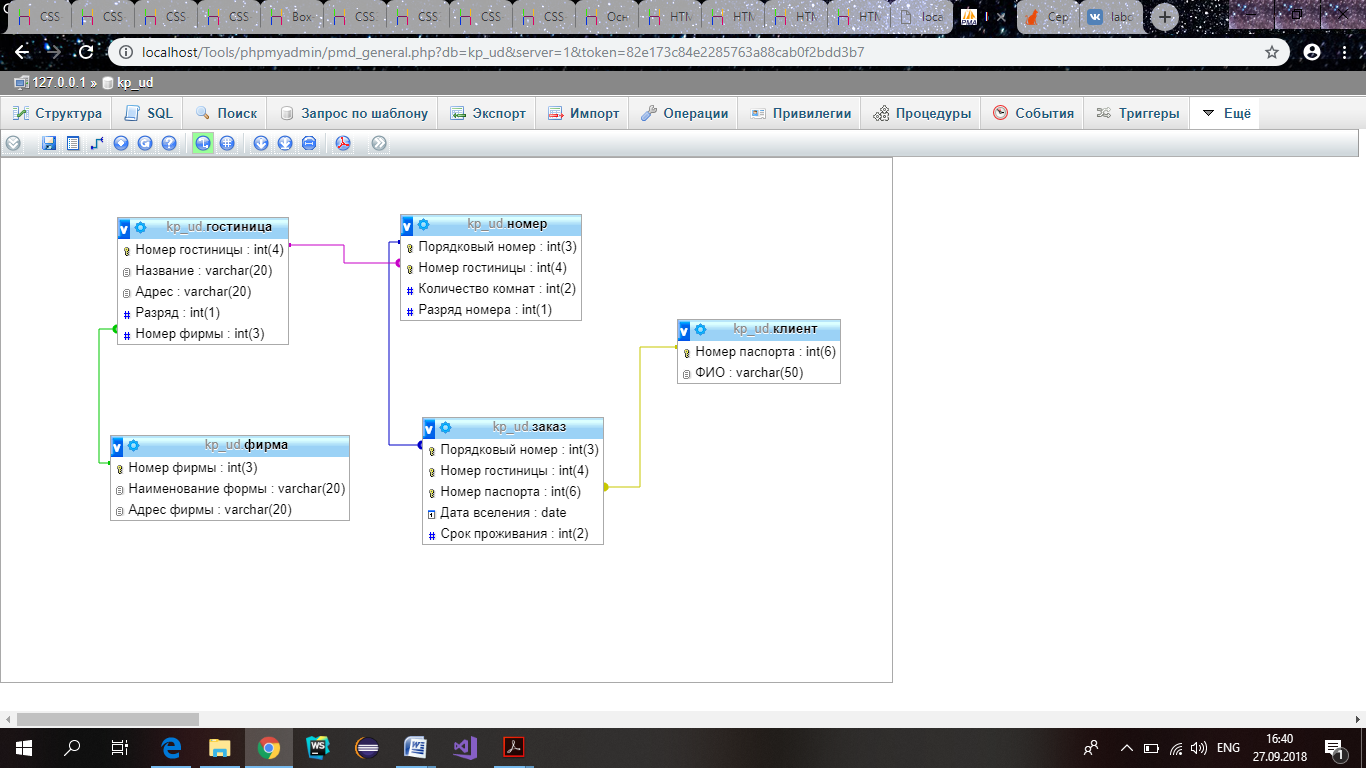


Рисунок 3.1 – Связи в таблице

1. Добавление записей в таблицу

INSERT INTO `kp\_ud`.`клиент` (`Номер паспорта`, `ФИО`) VALUES ('123459', 'Sirota Marina Romanovna');

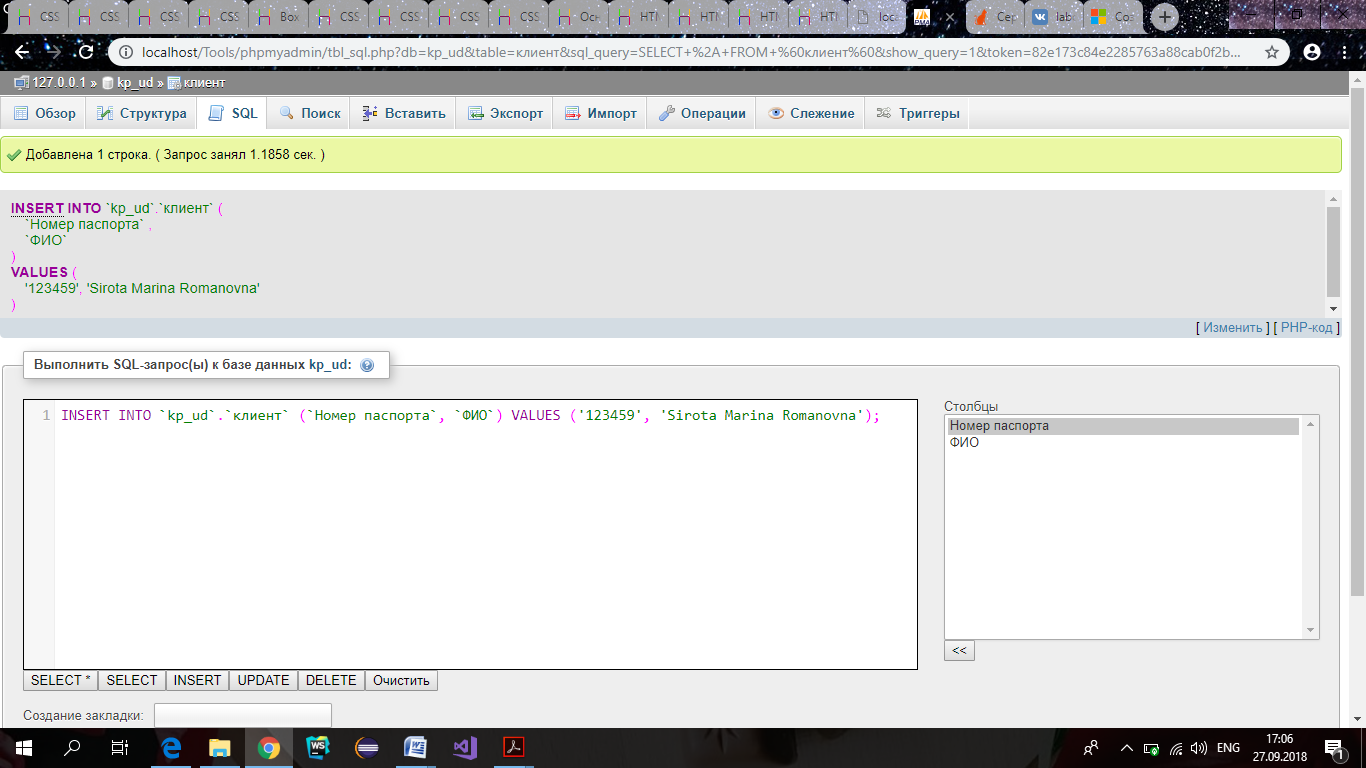


Рисунок 3.2 – Добавление записей в таблицу

1. Демонстрация простых вычислений в запросе:

SELECT `Номер гостиницы`, `Разряд`\*2 FROM `гостиница` WHERE `Разряд`='3'

*(Выбрать записи и удвоить разряд гостиницам с разрядом = 3 из таблицы Гостиница)*

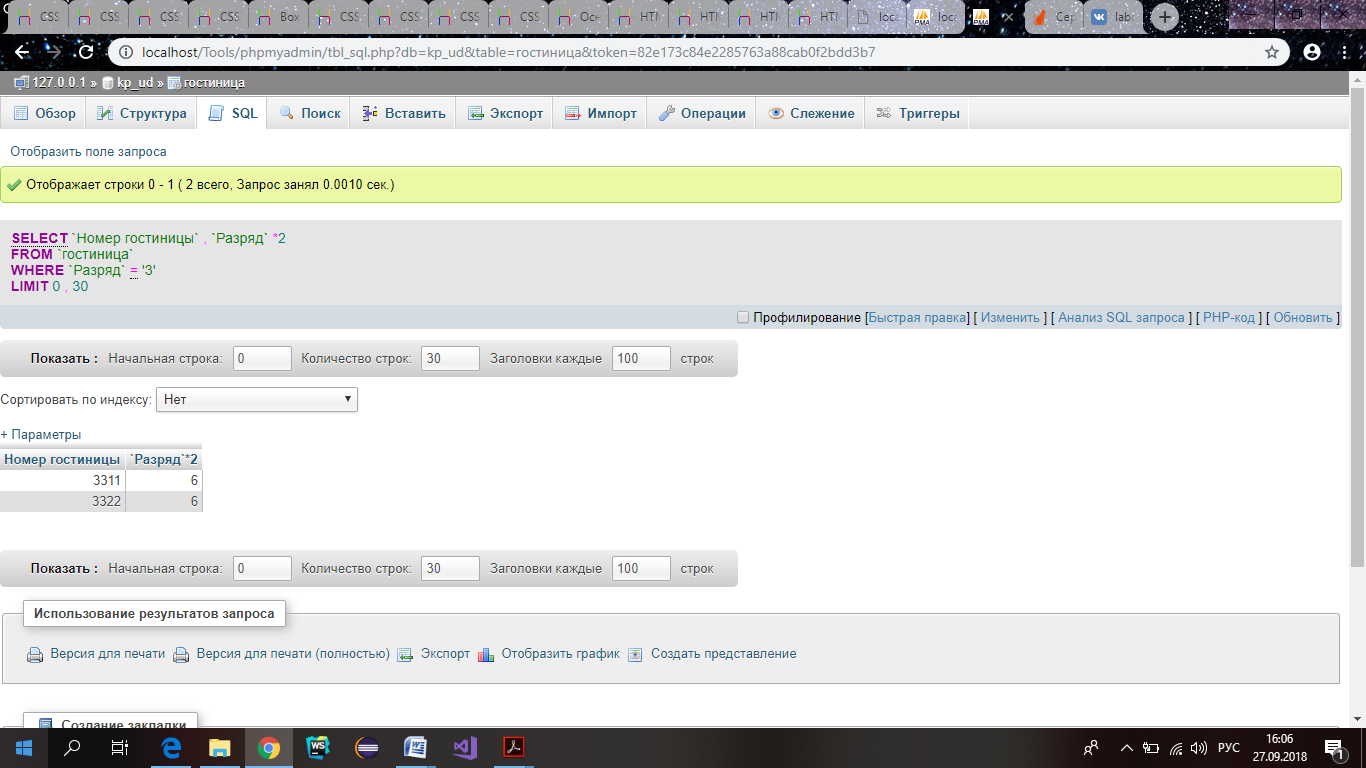


Рисунок 3.3 – Простые вычисления в запросе

1. Использование простого вычисления, как параметра агрегатной функции в запросе

SELECT min(`Срок проживания`+3) FROM `заказ`

*(Выбрать минимальный срок проживания из таблицы Заказ и увеличить на 3)*

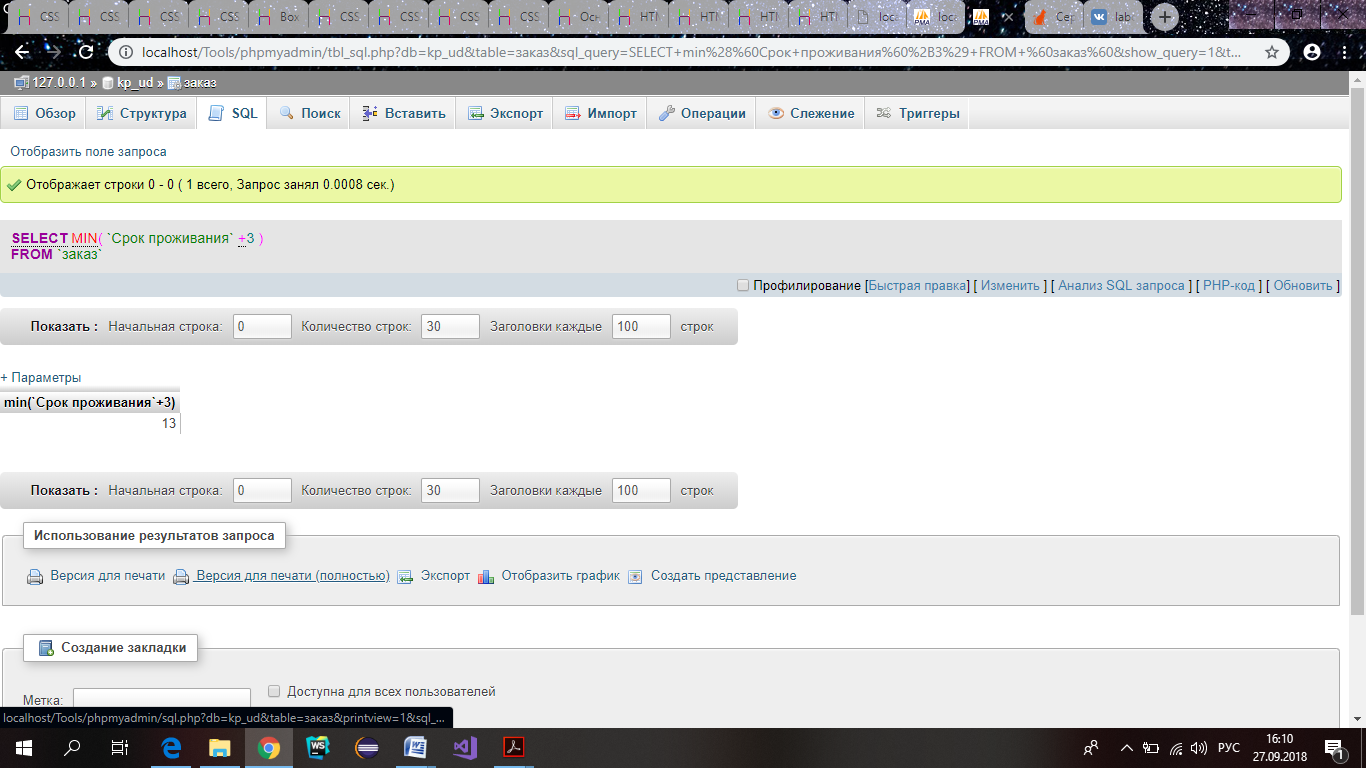
**

Рисунок 3.4 – Простые вычисления, как параметр агрегатной функции в запросе

1. Демонстрация работы функции HAVING

SELECT `Дата вселения`, `Номер паспорта`,`Срок проживания` FROM `заказ` GROUP BY `Дата вселения` HAVING (`Срок проживания`)<25

*(*Выбрать записи из таблицы *Заказ* с полями *Дата вселения, Номер паспорта* и *Срок проживания, у* которых значение поля *Срок проживания <25* и сгруппировать по полю *Дата проживания)*

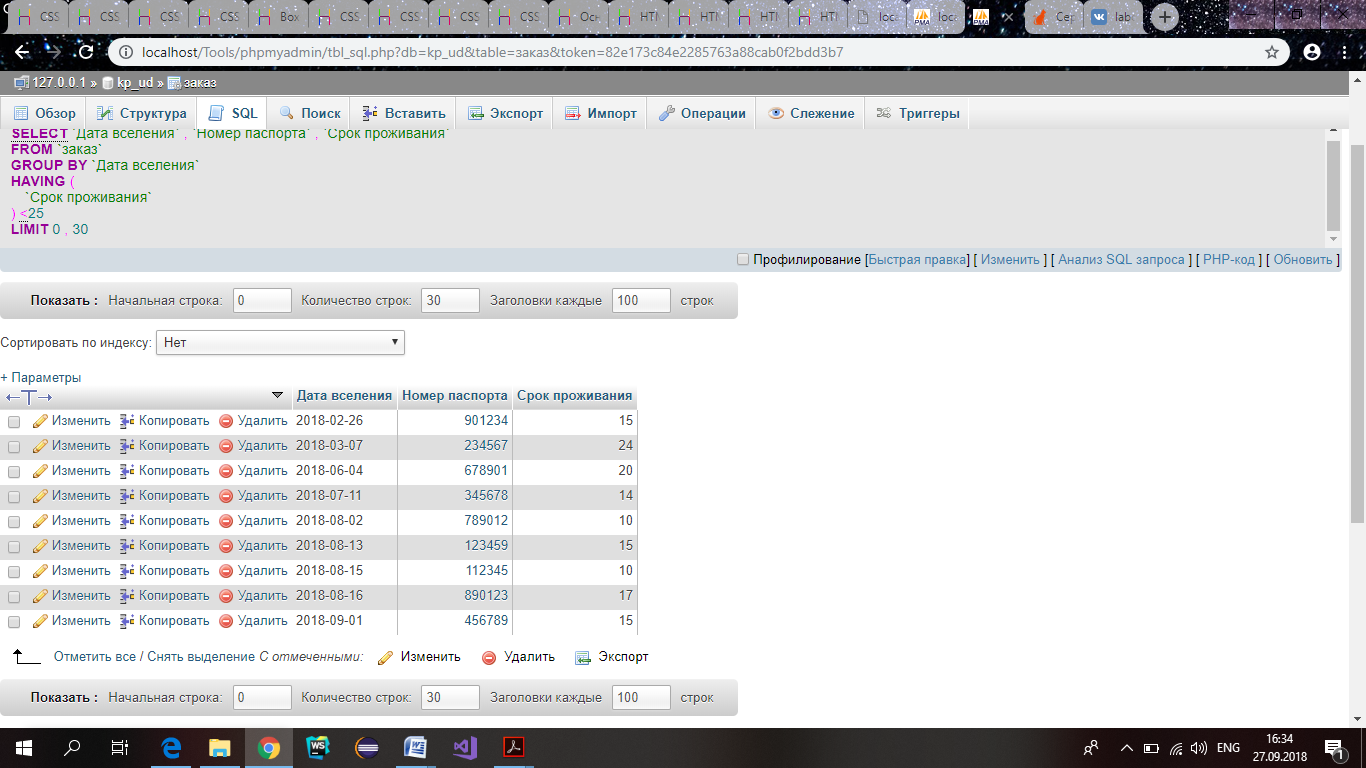


Рисунок 3.5 – Демонстрация работы функции HAVING

1. Демонстрация работы функции GROUP BY

SELECT `Дата вселения`, `Номер паспорта` FROM `заказ` GROUP BY `Дата вселения`

*(*Выбрать записи из таблицы *Заказ* с полями *Дата вселения, Номер паспорта* и сгруппировать по полю *Дата проживания)*

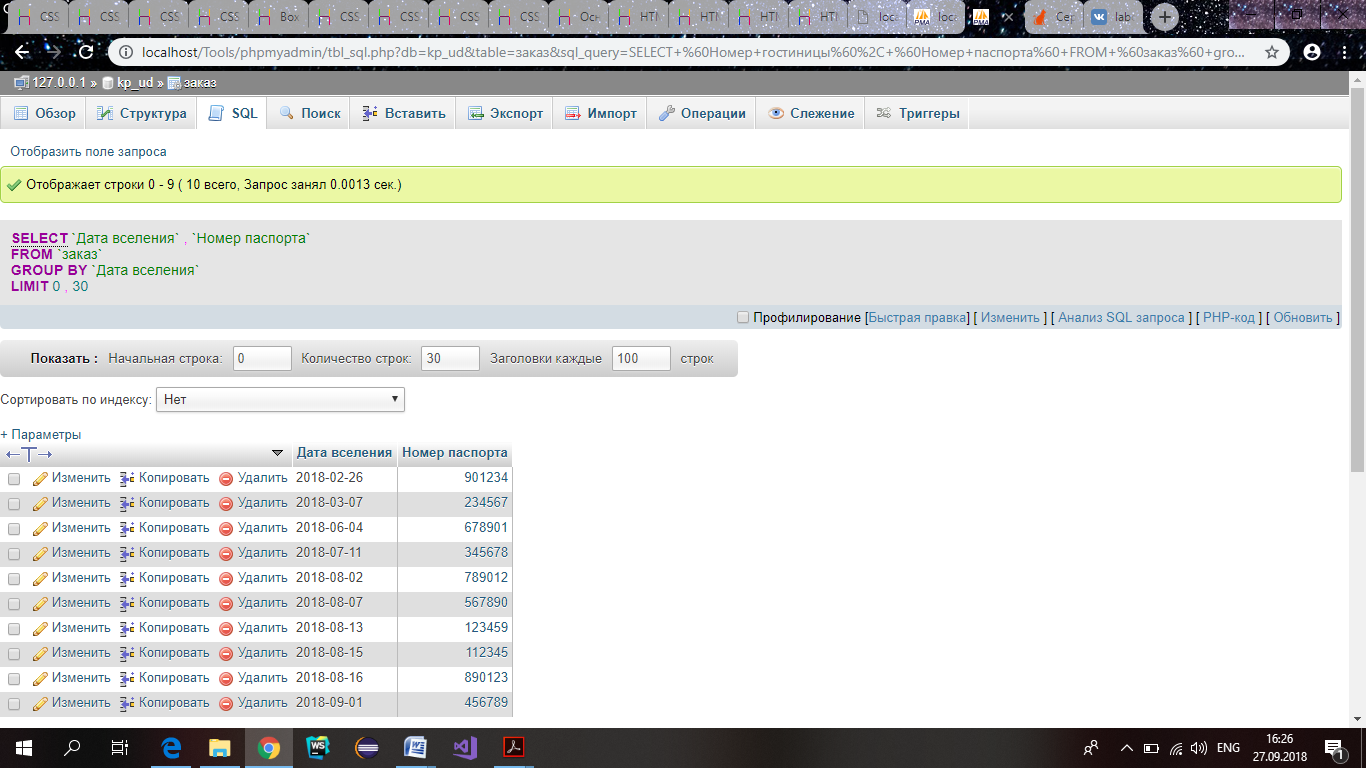
**

Рисунок 3.6 – Демонстрация работы функции GROUP BY

1. Селекция и соединение в одном запросе

Вывести *номер паспорта* и *срок проживания* из таблицы *заказ* и *название* *гостиницы* из таблицы *гостиница* для записей, у которых совпадают номера гостиниц.

SQL:

SELECT `Номер паспорта` , `заказ`.`Срок проживания`, `гостиница`.`Название` from `гостиница`,`заказ` where (`заказ`.`номер гостиницы`=`гостиница`.`номер гостиницы`)

Вывести номер паспорта, срок проживания и название гостиницы.

PA:

R = гостиница >< заказ

номер гостиницы = номер гостиницы

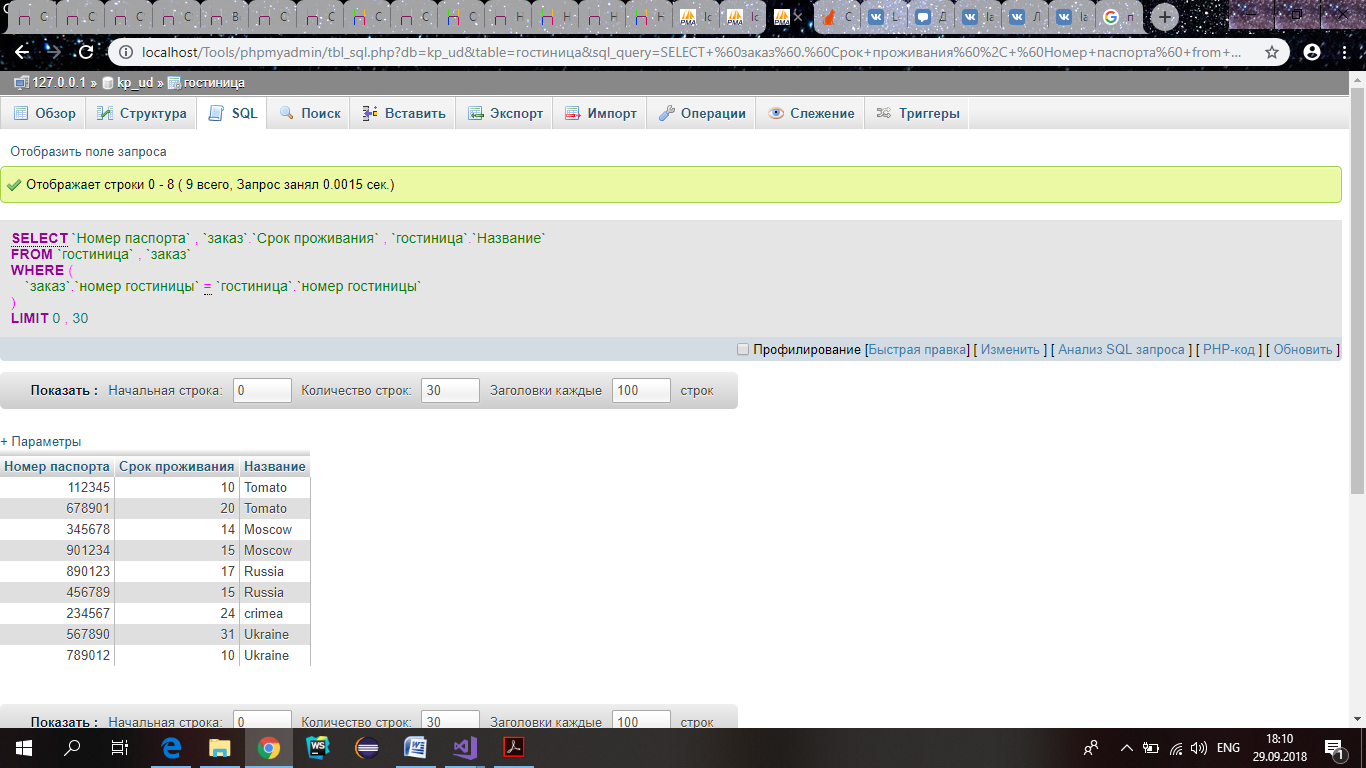


Рисунок 3.7 – Селекция и соединение в одном запросе

1. Запрос, использующий операции проекции, объединения и конъюнкции (в одном запросе)

Вывести ФИО тех клиентов, которые отдыхали меньше 20 дней и заселились раньше 1 мая 2018 года и тех, которые отдыхали больше 15 дней и заселились позже 21 июня 2018 года.

SQL:  
SELECT distinct `клиент`.`ФИО` FROM `клиент`, `заказ` WHERE(`заказ`.`срок проживания`)<20 and (`заказ`. `дата вселения`)<2018-05-01 and (`клиент`.`номер паспорта`=`заказ`.`номер паспорта`)

union

SELECT distinct `клиент`.`ФИО` FROM `клиент`, `заказ` WHERE(`заказ`.`срок проживания`)>15 and (`заказ`. `дата вселения`)>2018-06-21 and (`клиент`.`номер паспорта`=`заказ`.`номер паспорта`)

Реляционная алгебра:

R = R1 ∩ R2

R1 = π ФИО σ Срок проживания < 20 and Дата вселения < 2018-05-01

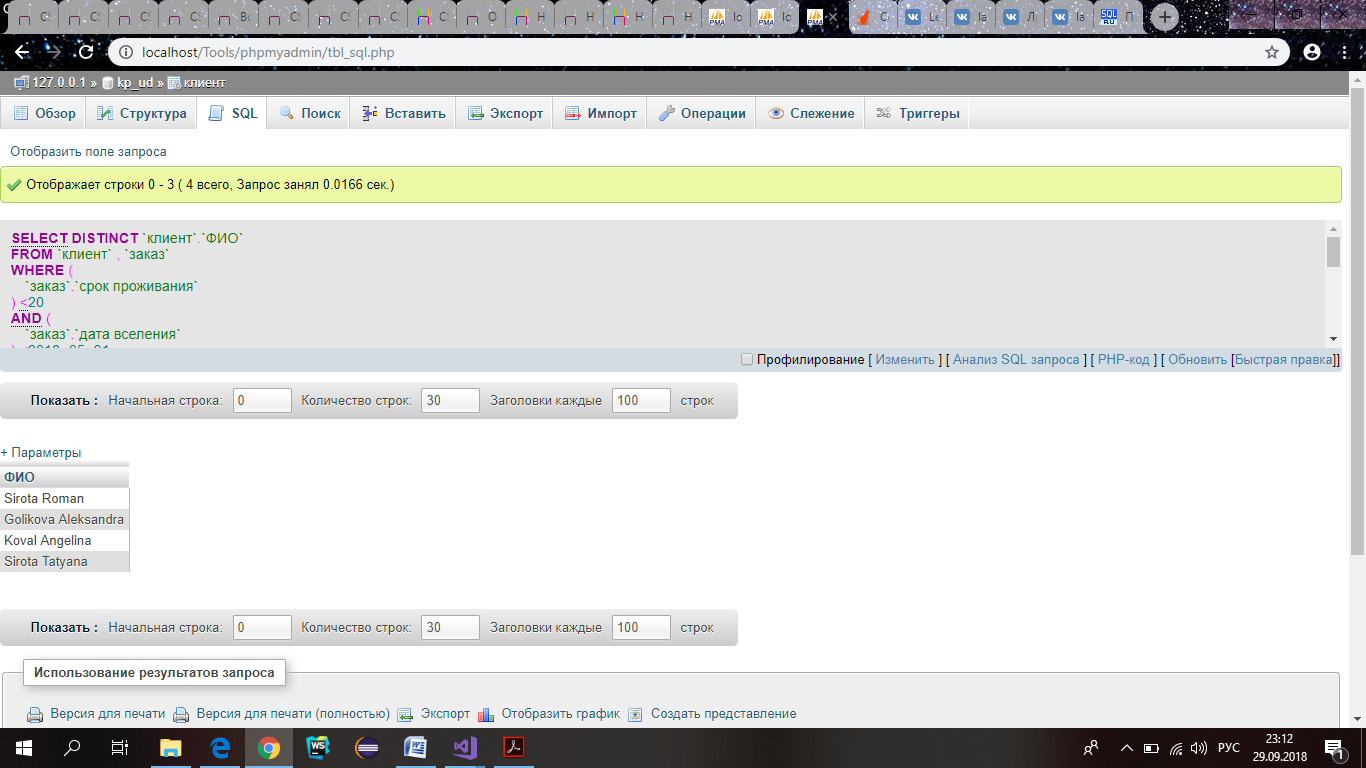
R2 = π ФИО σ Срок проживания > 15 and Дата вселения > 2018-06-21

Рисунок 3.8 – Проекция, объединение и конъюнкция в одном запросе

1. Запрос, использующий операции проекции и деления (в одном запросе).

Выбрать номера фирм, осуществляющих заселение в гостиницы со всеми возможными разрядами.

SQL:

SELECT DISTINCT `Номер фирмы`

FROM гостиница A

WHERE EXISTS

(SELECT `Номер фирмы`

FROM ФИРМА

WHERE NOT EXISTS

(SELECT `Номер фирмы`

FROM гостиница B

WHERE B.`Номер фирмы`=`ФИРМА`.`Номер фирмы` AND A.`Разряд` = B.`Разряд`))

Реляционная алгебра:

PA:

R=Гостиница/Фирма

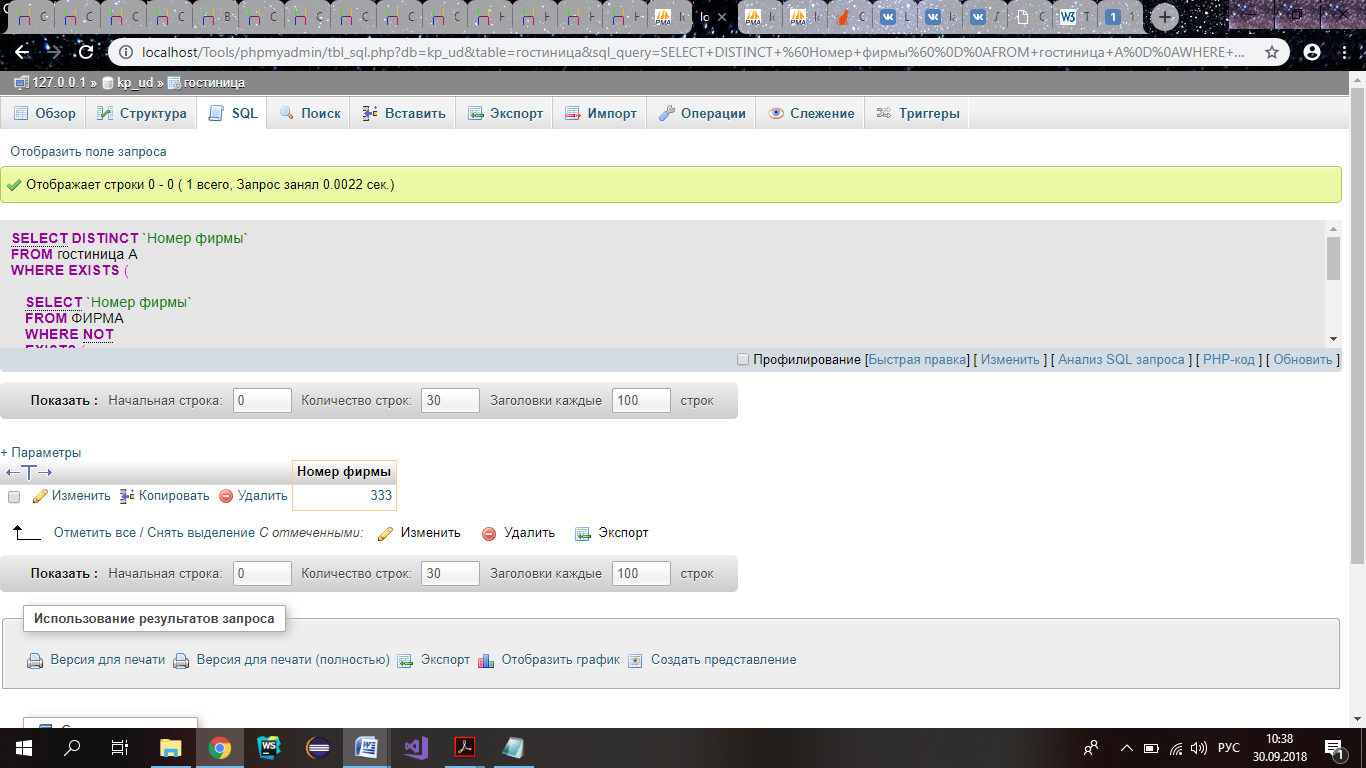


Рисунок 3.9 – Проекция и деление в одном запросе

1. Запрос, использующий операции вычитания и дизъюнкции (в одном запросе).

Выдать номер паспортов тех клиентов, которые отдыхали либо меньше 15 дней, либо больше 25.

SQL:

SELECT `Номер паспорта`FROM `заказ`

WHERE NOT EXISTS

(SELECT `Номер паспорта` FROM `заказ` WHERE `Срок проживания` between 15 and 25 )

SELECT `Номер паспорта`FROM `заказ`

WHERE `Номер паспорта` in

(SELECT `Номер паспорта` FROM `заказ` WHERE `Срок проживания` between 15 and 25 )

Реляционная алгебра:

РА = R1 – R2

R1 = π Номер паспорта

R2 = π Номер паспорта, σ срок проживания (15;25)

ВЫВОДЫ

В ходе первой лабораторной работы по дисциплине «Управление данными» были изучены основы реляционной алгебры как базового средства манипулирования.

Были получены навыки представления запросов как на реляционной алгебре, так и на SQL.