Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**Основы DМL-запросов в PostgreSQL**

Лабораторная работа №2

по курсу «Базы данных»

Выполнил студент группы ИВТб-31 /Категов А.Д/ Проверил преподаватель /Клюкин В.Л./

Киров 2024

**Задание на лабораторную работу:**

При выполнении работы следует использовать базу данных, созданную в лабораторной работе №1. Ниже приведены задания, которые необходимо выполнить в ходе лабораторной работы:

* Создать и выполнить SQL-скрипт, который будет заполнять таблицы данными. Нужно добавить не менее 3-5 строк в каждую таблицу.
* Создать представления для нескольких таблиц, в которых собираются данные из самой таблицы и других, на которые она ссылается. Среди представлений обязательно должно быть представление для таблицы, которая используется для связи «многие-ко-многим». Выборка из любого представления должна давать полную и осмысленную информацию по сущностям. Хотя бы одно из представлений должно быть сделано с использованием соединений (join) в запросе.
* Для любой таблицы, содержащей столбец с числовыми данными, создать представление следующего вида, отражающее информацию по этому столбцу (в представлении ровно 3 столбца и 4 строки):

|  |  |  |
| --- | --- | --- |
| Минимальное значение | <значение> | <id записи с минимальным значением> |
| Максимальное значение | <значение> | <id записи с максимальным значением> |
| Среднее значение | <значение> | null |
| Сумма значений | <значение> | null |

**Ход работы**

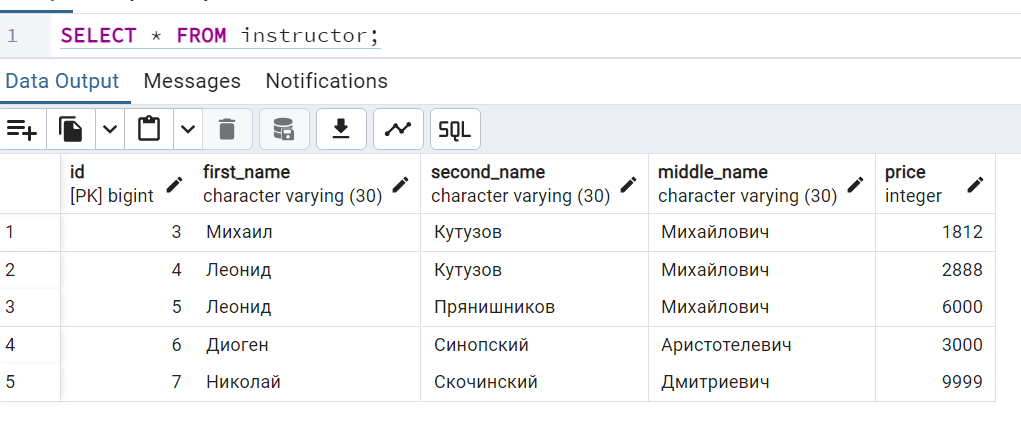
****

Рисунок 1 – Заполненная таблица инструкторов.

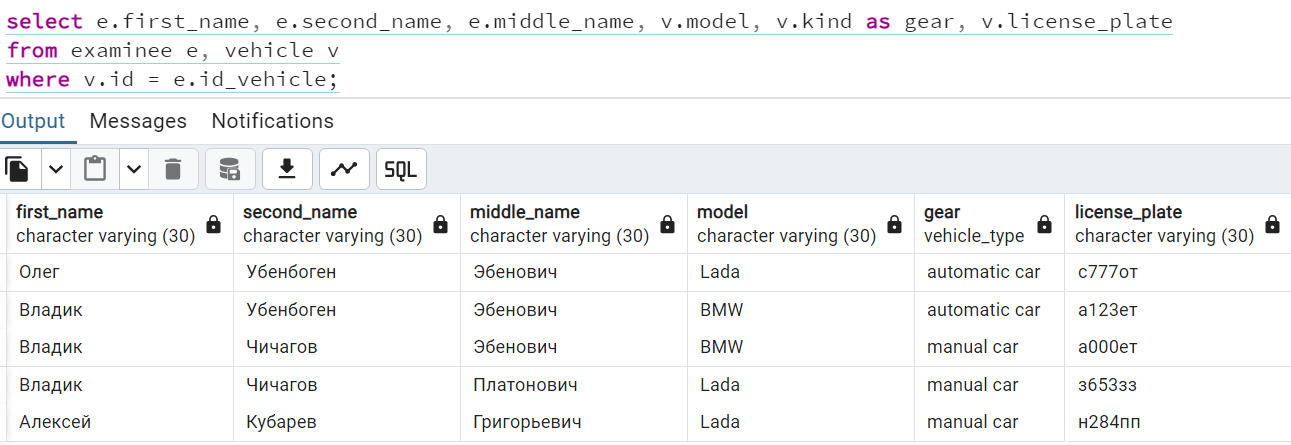
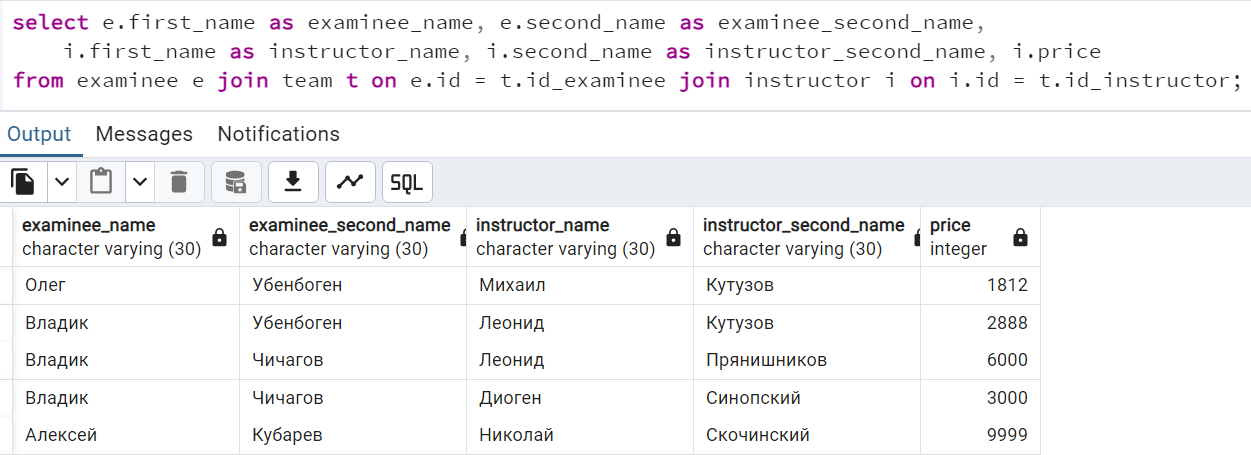
****

Рисунок 2 – Пример команды SELECT для экзаменуемых

****

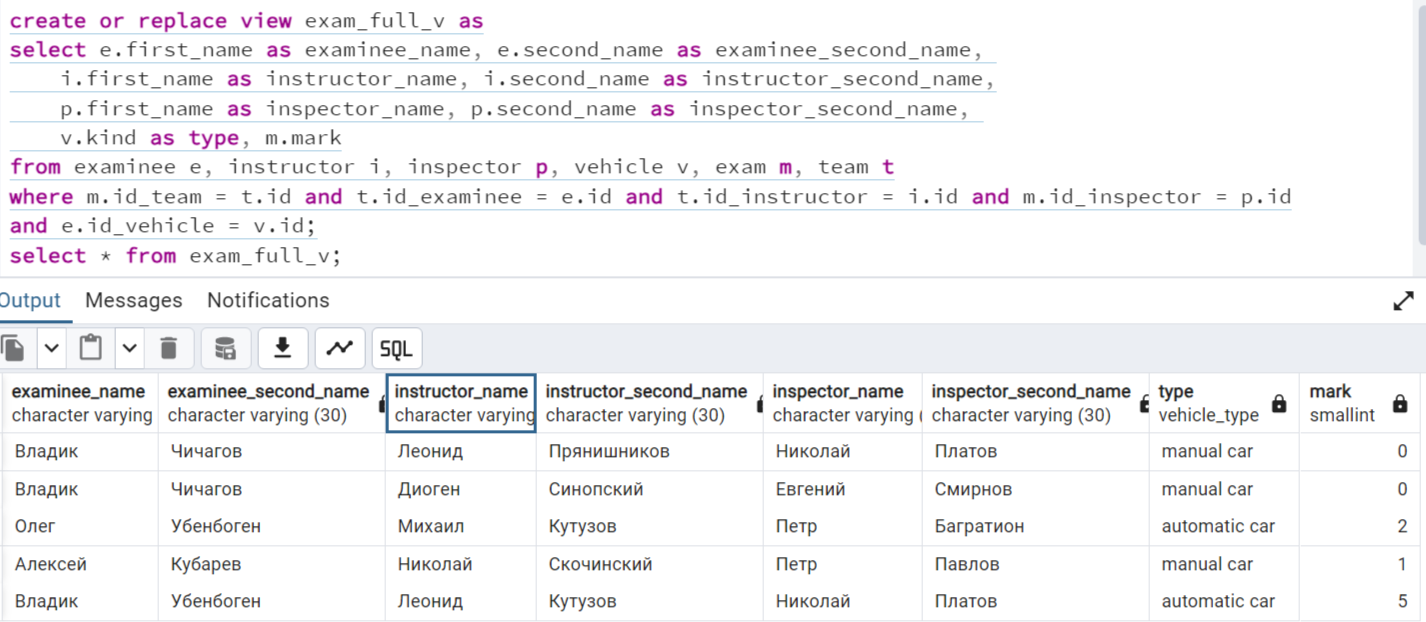
****Рисунок 3 – Пример JOIN и SELECT для групп

Рисунок 4 – Представление-VIEW для экзамена

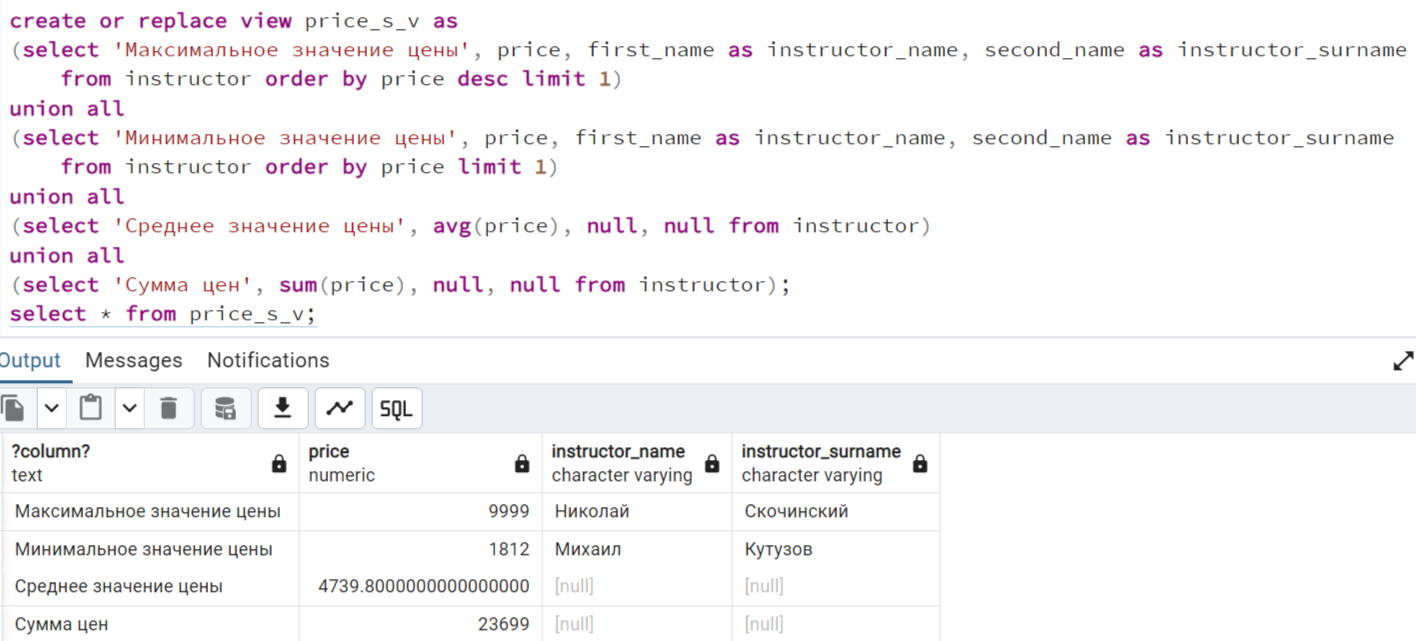
****

Рисунок 5 – Представление с агрегатными функциями для цен

**Листинг программы**

do language plpgsql $$

declare

id\_vehicle bigint;

id\_instructor bigint;

id\_examinee bigint;

id\_team bigint;

id\_inspector bigint;

begin

insert into vehicle(model, license\_plate, kind)

values('Lada', 'н284пп', 'manual car')

returning id into id\_vehicle;

insert into examinee(first\_name, second\_name, middle\_name, id\_vehicle)

values('Алексей', 'Кубарев', 'Григорьевич', id\_vehicle)

returning id into id\_examinee;

insert into instructor(first\_name, second\_name, middle\_name, price)

values('Николай', 'Скочинский', 'Дмитриевич', 9999)

returning id into id\_instructor;

insert into team(id\_instructor, id\_examinee)

values(id\_instructor, id\_examinee)

returning id into id\_team;

insert into inspector(first\_name, second\_name, middle\_name)

values('Петр', 'Павлов', 'Павлович')

returning id into id\_inspector;

insert into exam(mark, id\_team, id\_inspector)

values(0, id\_team, id\_inspector);

end;

$$

select e.first\_name, e.second\_name, e.middle\_name, v.model, v.kind as gear, v.license\_plate

from examinee e, vehicle v

where v.id = e.id\_vehicle;

select e.first\_name as examinee\_name, e.second\_name as examinee\_second\_name,

i.first\_name as instructor\_name, i.second\_name as instructor\_second\_name, i.price

from examinee e join team t on e.id = t.id\_examinee join instructor i on i.id = t.id\_instructor;

create or replace view exam\_full\_v as

select e.first\_name as examinee\_name, e.second\_name as examinee\_second\_name,

i.first\_name as instructor\_name, i.second\_name as instructor\_second\_name,

p.first\_name as inspector\_name, p.second\_name as inspector\_second\_name,

v.kind as type, m.mark

from examinee e, instructor i, inspector p, vehicle v, exam m, team t

where m.id\_team = t.id and t.id\_examinee = e.id and t.id\_instructor = i.id and m.id\_inspector = p.id

and e.id\_vehicle = v.id;

select \* from exam\_full\_v;

create or replace view price\_s\_v as

(select 'Максимальное значение цены', price, first\_name as instructor\_name, second\_name as instructor\_surname

from instructor order by price desc limit 1)

union all

(select 'Минимальное значение цены', price, first\_name as instructor\_name, second\_name as instructor\_surname

from instructor order by price limit 1)

union all

(select 'Среднее значение цены', avg(price), null, null from instructor)

union all

(select 'Сумма цен', sum(price), null, null from instructor);

select \* from price\_s\_v;

**Вывод:** в ходе данной лабораторной работы были изучены основы DML-запросов в PostgreSQL. В процессе работы были заполнены таблицы, созданные в первой лабораторной работе и изучены агрегатные функции, а также представление view. На основании всего сказанного выше можно сделать вывод о том, что цель достигнута, необходимые знания получены, задание выполнено верно, что доказывается полученными результатами.