Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №5

Реализация SQL-запросов на выборку данных с использованием подзапросов, агрегатных функций, группировки и операций над множествами

Студент: К.В. Горбачевский

Преподаватель: Д.В. Куприянова

МИНСК 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc162857018)

[1 SQL-ЗАПРОСЫ 4](#_Toc162857019)

[1.1 COUNT 4](#_Toc162857020)

[1.2 COUNT (DISTINCT) 5](#_Toc162857021)

[1.3 MAX 6](#_Toc162857022)

[1.4 MAX 7](#_Toc162857023)

[1.5 SUM 8](#_Toc162857024)

[1.6 SUM (DISTINCT) 9](#_Toc162857025)

[1.7 GROUP BY 10](#_Toc162857026)

[1.8 GROUP BY (HAVING) 11](#_Toc162857027)

[1.9 HAVING (AVG) 12](#_Toc162857028)

[1.10 UNION (NOT) 13](#_Toc162857029)

[1.11 UNION (ANY) 14](#_Toc162857030)

[1.12 INTERSECT 15](#_Toc162857031)

[1.13 INTERSECT (WHERE) 16](#_Toc162857032)

[1.14 EXCEPT (EXISTS) 17](#_Toc162857033)

[1.15 EXCEPT (ALL) 18](#_Toc162857034)

[ЗАКЛЮЧЕНИЕ 19](#_Toc162857035)

# ВВЕДЕНИЕ

В лабораторной работе выполняется создание запросов на выборку данных на языке SQL с использованием подзапросов, агрегатных функций, а также группировки данных (предложение GROUP BY оператора SELECT) и операций над множествами (UNION, INTERSECT, MINUS).

1) Получить у преподавателя задания по вашей собственной схеме данных, созданной в лабораторной работе №2 и реализованной в виде таблиц в СУБД в лабораторной работе №3. Создать запросы по заданиям (по одному запросу на каждое задание).

2) Правила выполнения заданий:

для каждого задания создать реализацию в виде одного оператора SQL **SELECT**, в котором можно использовать подзапросы и группировку данных;

обратить внимание, что использование скалярных (особенно соотнесенных !) подзапросов в предложении **SELECT** следует ограничить, т.к. они ухудшают производительность и анализ запроса, поэтому, если запрос затрагивает несколько таблиц, то сначала надо собрать данные с помощью соединения данных таблиц, и только потом выполнять их обработку (например, группировать);

перед запуском запроса на выполнение, изучить данные в используемых запросом таблицах, и если требуется добавить новые данные, чтобы результат выборки не был пустым;

выполнить запрос и проанализировать его результат – если есть расхождения между изученными данными и результатом запроса, то есть повод задуматься о проверке правильности выполнения этого задания.

3) Оформить отчет.

# SQL-ЗАПРОСЫ

* 1. COUNT

Вывести с количеством записей артистов, с информацией о комнате и записи.



Рисунок 1.1.0 – Исходная таблица

На рисунке 1.1.1 показан код SQL-скрипта и результат его работы

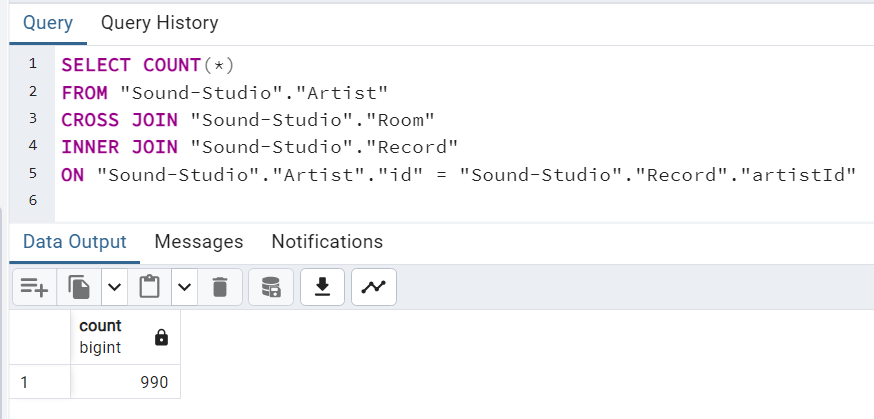


Рисунок 1.1.1 – SQL-cкрипт и результат выполнения

* 1. COUNT (DISTINCT)

Вывести количество записей в объединеной таблице Room, c уникальным значением площади, артистов в жанре Рок. Объединить с таблице Record



Рисунок 1.2.0 – Исходная таблица

На рисунке 1.2.1 показан код SQL-скрипта и результат его работы

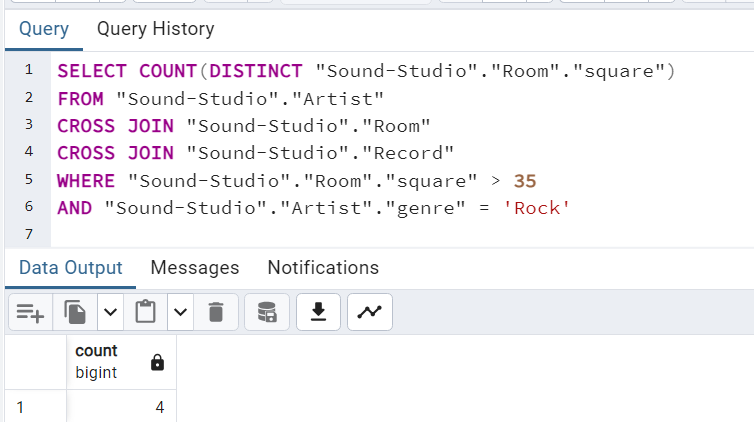


Рисунок 1.2.1 – SQL-cкрипт и результат выполнения

* 1. MAX

Вывести максимальное количество участников в составе группы, объединить с таблицей Record и Instrument.



Рисунок 1.3.0 – Исходная таблица

На рисунке 1.3.1 показан код SQL-скрипта и результат его работы

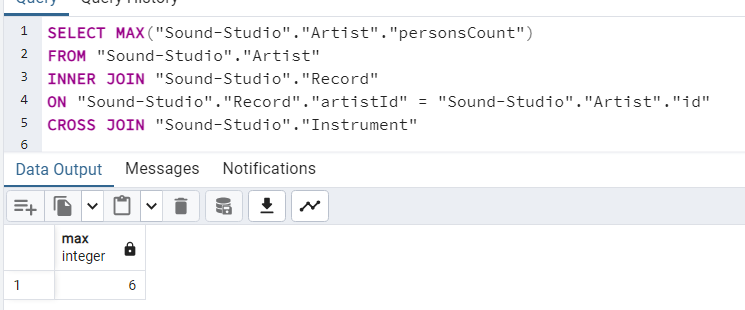


Рисунок 1.3.1 – SQL-cкрипт и результат выполнения

* 1. MAX

Вывести максимальную цену за запись, объединить с сотрудником и аристом

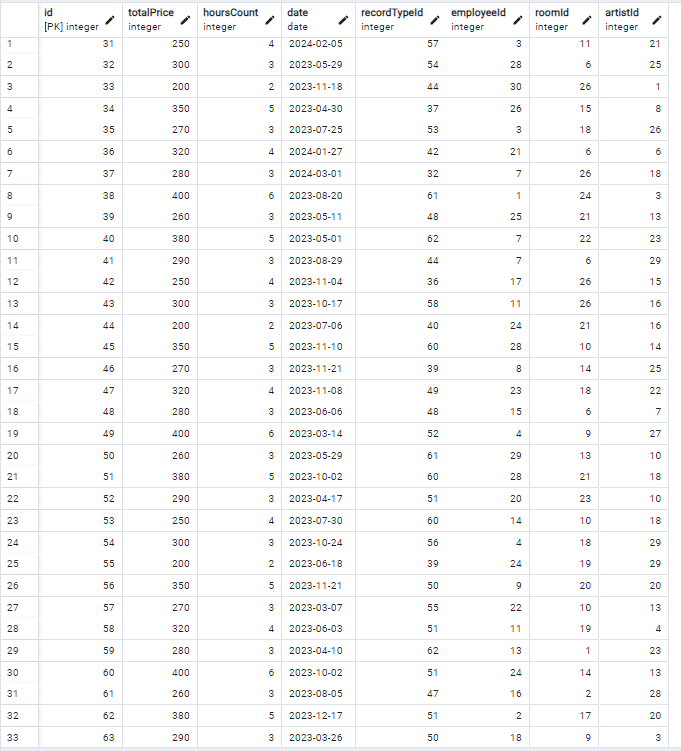


Рисунок 1.4.0 – Исходная таблица

На рисунке 1.4.1 показан код SQL-скрипта и результат его работы

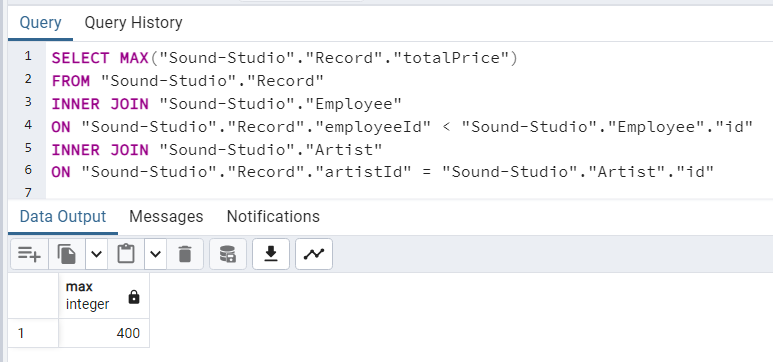


Рисунок 1.4.1 – SQL-cкрипт и результат выполнения

* 1. SUM

Вывести сумму площадей всех комнат, объединить с сотрудниками, работающими во вторую смену и тип записи – вокал.

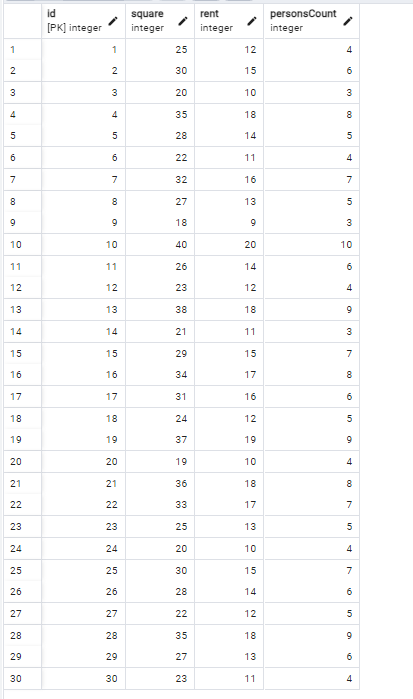


Рисунок 1.5.0 – Исходная таблица

На рисунке 1.5.1 показан код SQL-скрипта и результат его работы

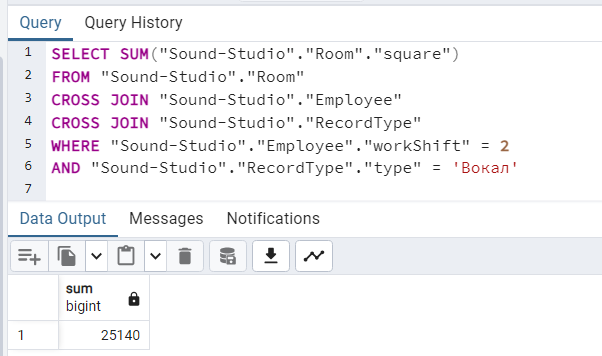


Рисунок 1.5.1 – SQL-cкрипт и результат выполнения

* 1. SUM (DISTINCT)

Вывести сумму уникальных цен записи, где количество человек в комнате не больше 4 и сотрудник, работающий в первую смену.

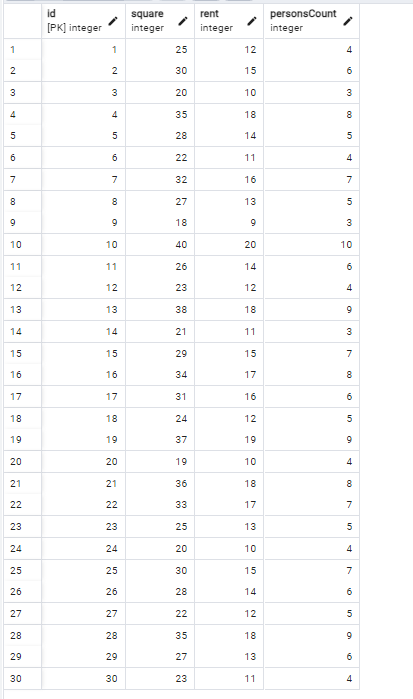


Рисунок 1.6.0 – Исходная таблица Room

На рисунке 1.6.1 показан код SQL-скрипта и результат его работы

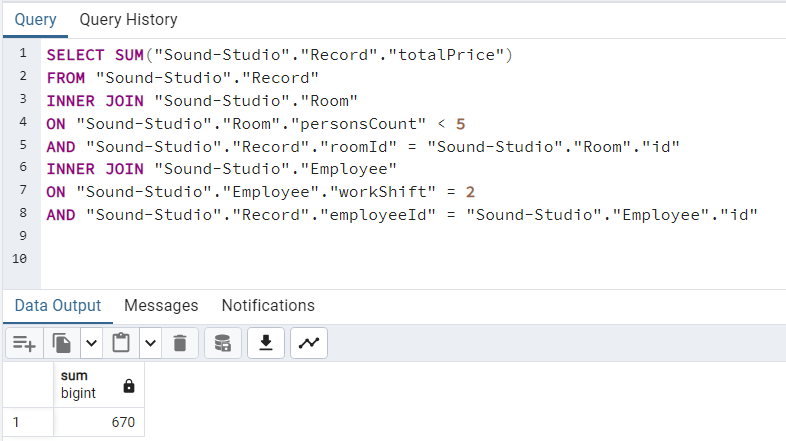


Рисунок 1.6.1 – SQL-cкрипт и результат выполнения

* 1. GROUP BY

Вывести сумму всех записей группы и объединить с типом записи



Рисунок 1.7.0 – Исходная таблица Artist

На рисунке 1.7.1 показан код SQL-скрипта и результат его работы

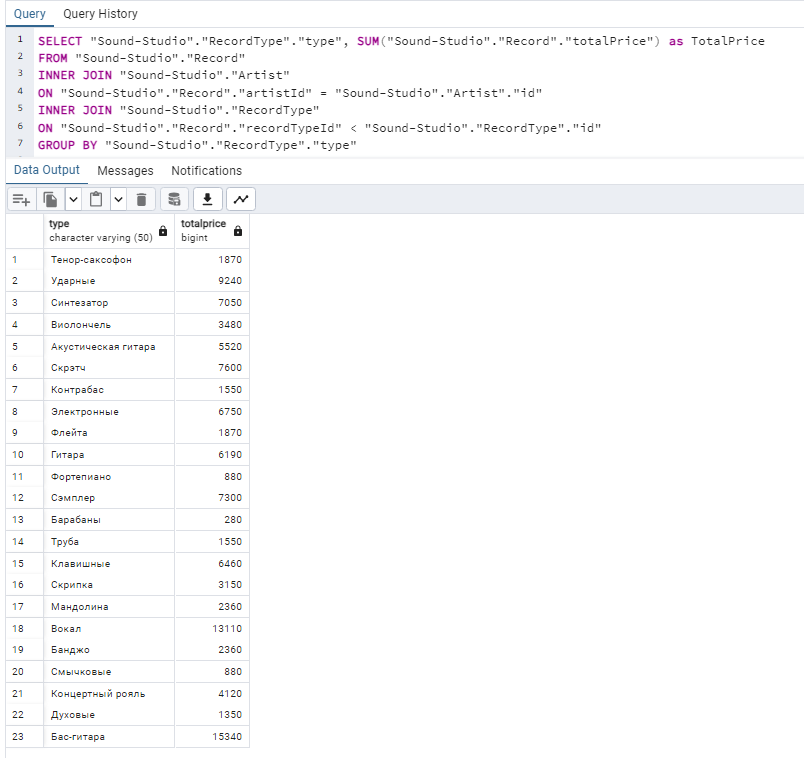


Рисунок 1.7.1 – SQL-cкрипт и результат выполнения

* 1. GROUP BY (HAVING)

Вывести запись артистов, c минимальным числом участников, объединить с комнатой, в которой будет происходить запись после 5 июля 2023 года



Рисунок 1.8.0 – Исходная таблица Employee

На рисунке 1.8.1 показан код SQL-скрипта и результат его работы

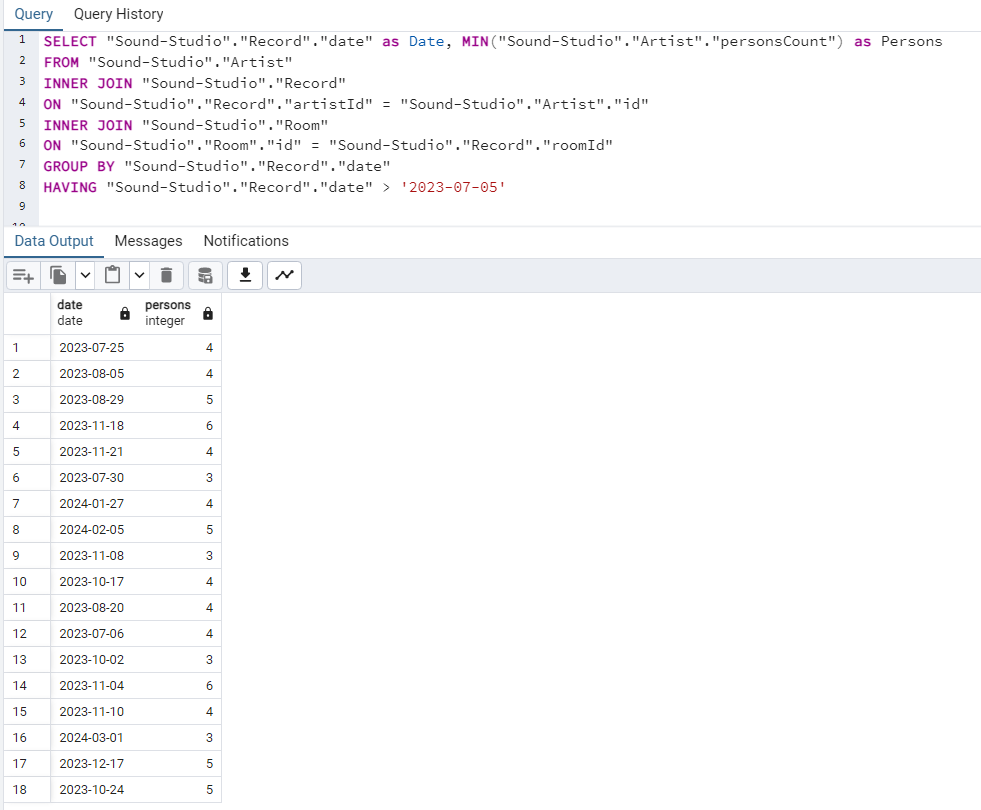


Рисунок 1.8.1 – SQL-cкрипт и результат выполнения

* 1. HAVING (AVG)

Вывести аристов в жанре Рок или Метал и среднюю цену записи, у которых тип записи – стерео. Сгрупировать по имени исполнителя в прямом порядке.

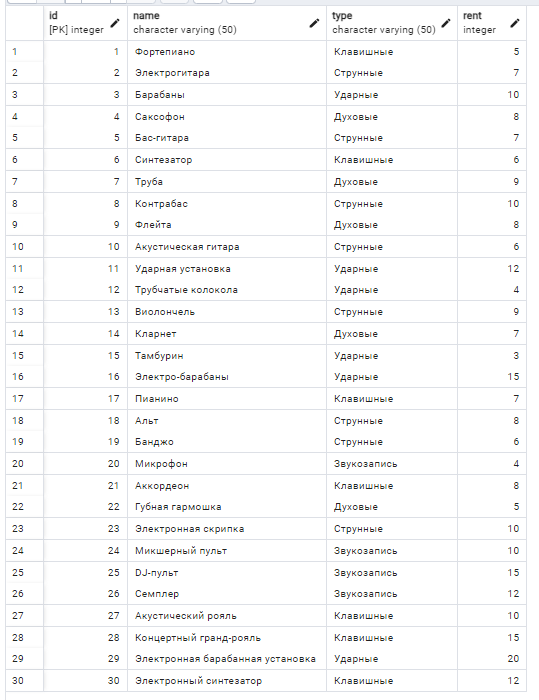


Рисунок 1.9.0 – Исходная таблица Instrument

На рисунке 1.9.1 показан код SQL-скрипта и результат его работы

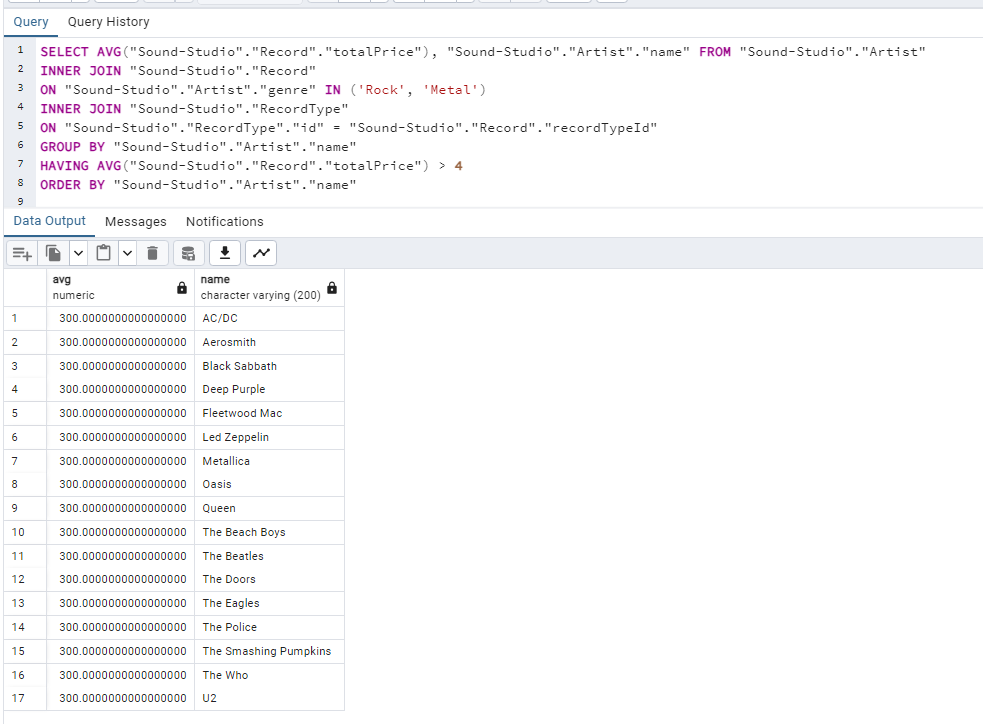


Рисунок 1.9.1 – SQL-cкрипт и результат выполнения

* 1. UNION (NOT)

Вывести артистов c односложным названием

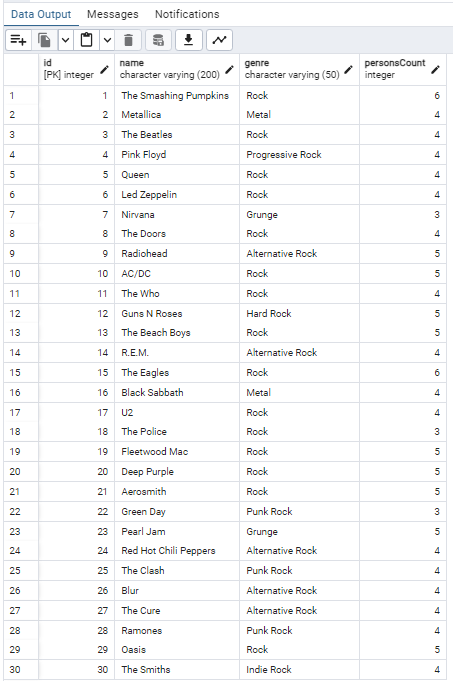


Рисунок 1.10.0 – Исходная таблица Artist

На рисунке 1.10.1 показан код SQL-скрипта и результат его работы

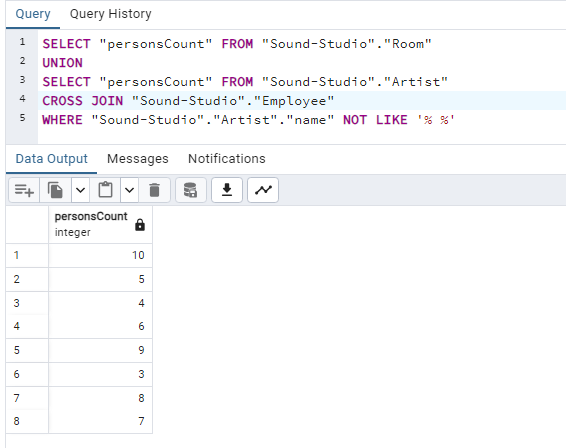


Рисунок 1.10.1 – SQL-cкрипт и результат выполнения

* 1. UNION (ANY)

Вывести запись, в которой сотрудник третьей смены записывает артиста с максимальным id

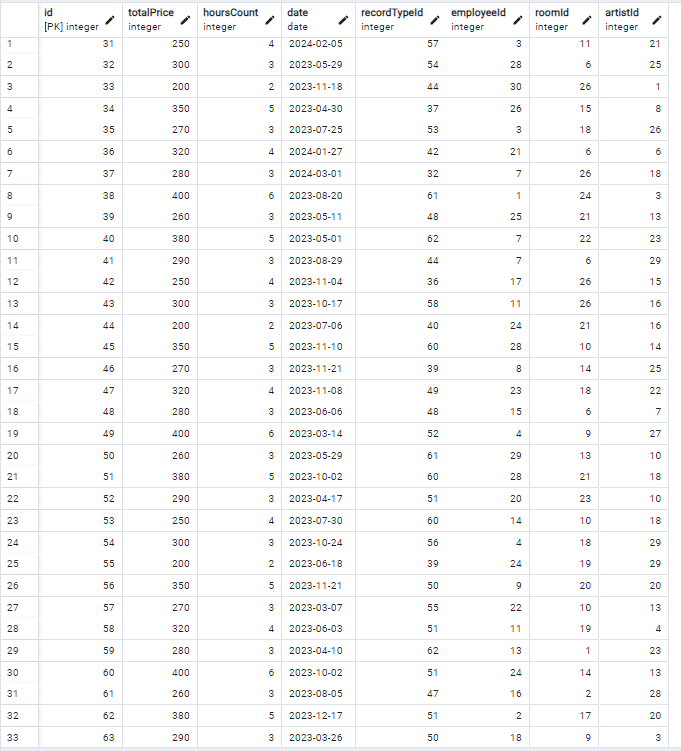


Рисунок 1.11.0 – Исходная таблица Record

На рисунке 1.11.1 показан код SQL-скрипта и результат его работы



Рисунок 1.11.1 – SQL-cкрипт и результат выполнения

* 1. INTERSECT

Вывести сотрудников, которые вместе с этим являются исполнителями.



Рисунок 1.12.0 – Исходная таблица Employee

На рисунке 1.12.1 показан код SQL-скрипта и результат его работы

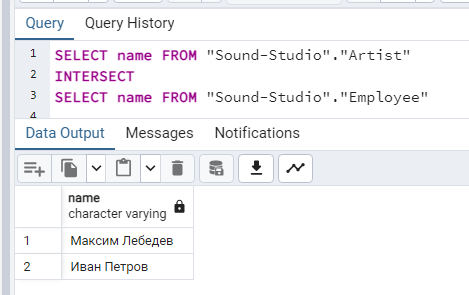


Рисунок 1.12.1 – SQL-cкрипт и результат выполнения

* 1. INTERSECT (WHERE)

Вывести тип записи и инструмент, который можно записать с ценой аренды менее 5.

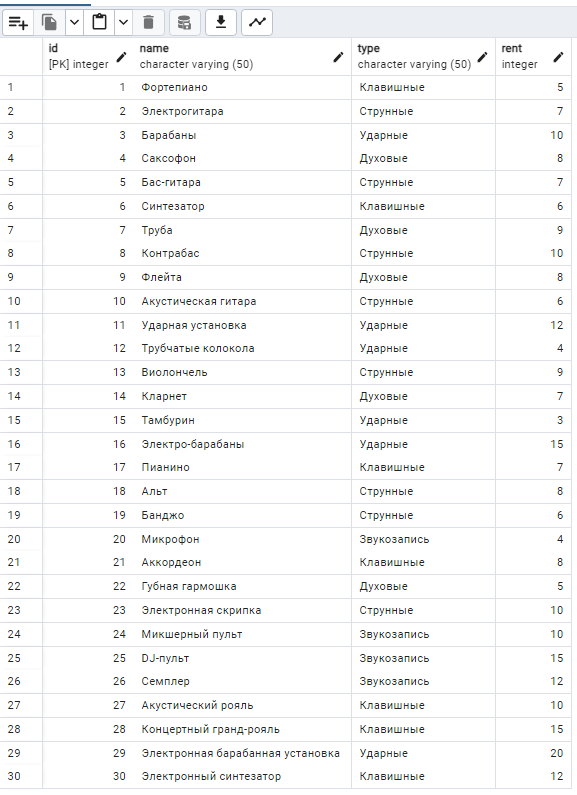


Рисунок 1.13.0 – Исходная таблица Instrument

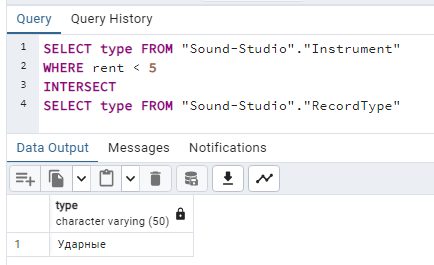


Рисунок 1.13.1 – SQL-cкрипт и результат выполнения

* 1. EXCEPT (EXISTS)

Вывести уникальные инструменты с такой ценой аренды, которая хотя бы раз соответсвует цене аренды комнаты.

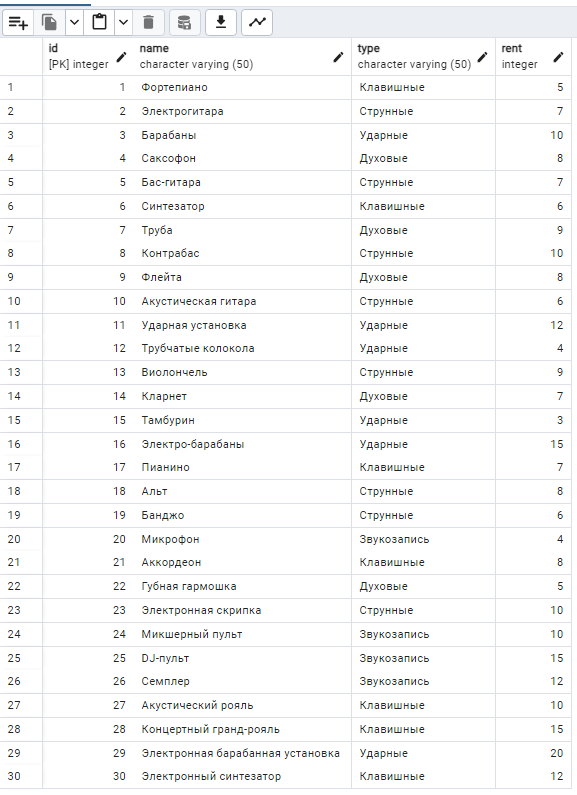


Рисунок 1.14.0 – Исходная таблица Instrument

На рисунке 1.14.1 показан код SQL-скрипта и результат его работы

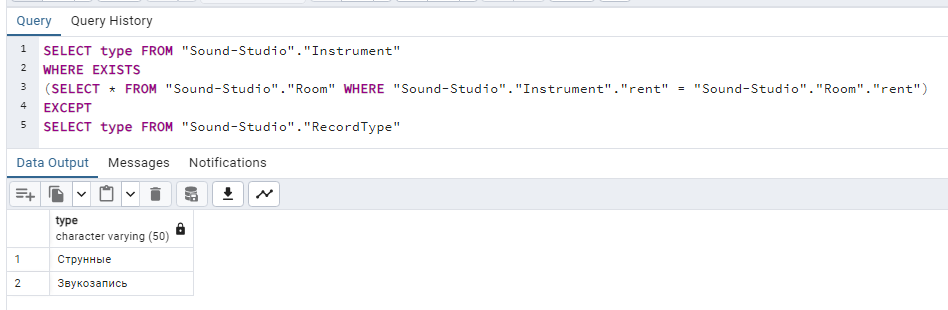


Рисунок 1.13.1 – SQL-cкрипт и результат выполнения

* 1. EXCEPT (ALL)

Вывести уникальную стоимость услуги, объединив таблицу с комнатой и записью. Объединить с инструментами, если стоимость аренды любой из комнат больше 5

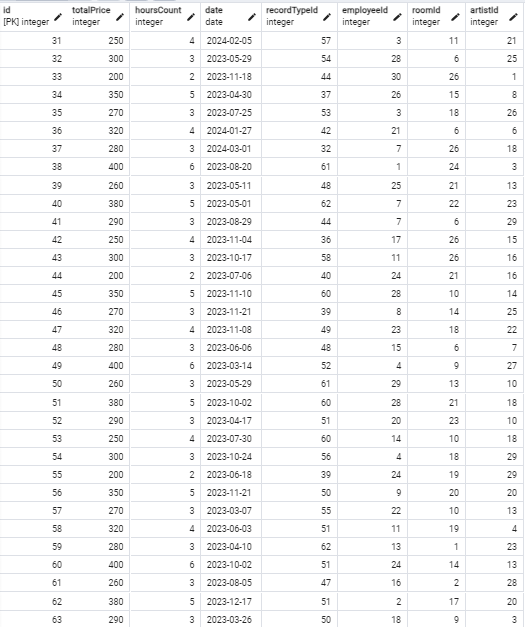


Рисунок 1.15.1 – Исходная таблица Record

На рисунке 1.15.1 показан код SQL-скрипта и результат его работы

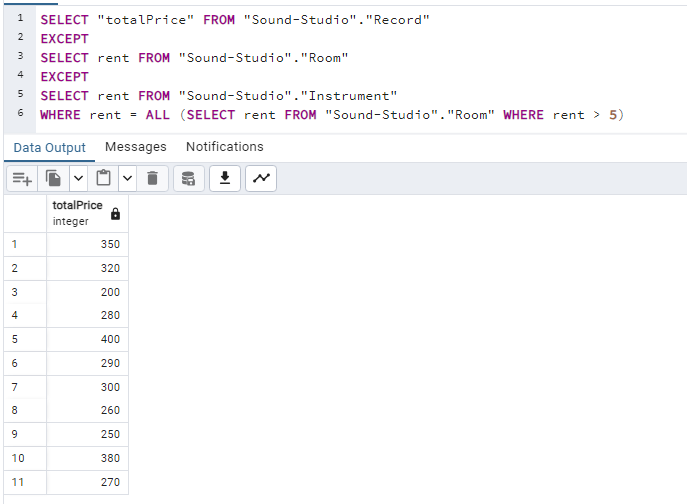


Рисунок 1.15.1 – SQL-cкрипт и результат выполнения

# ЗАКЛЮЧЕНИЕ

В ходе данной лабораторной работы были изучены дополнительные возможности SQL-запросов в Postgres на базе «PgAdmin 4».

Изучено формирование SQL-операторов для выборки данных из таблиц с использованием многотабличных запросов, подзапросов и секций WHERE, ORDER BY и других.