Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Тема «Репетиционная база»

Лабораторная работа №5

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

Студент: А.С. Бригадир

Преподаватель: Д.В. Куприянова

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc193793903)

[1 SQL-ЗАПРОСЫ 4](#_Toc193793904)

[1.1 Rehearsal\_points 4](#_Toc193793905)

[1.2 Rooms 5](#_Toc193793906)

[1.3 Service 7](#_Toc193793907)

[1.4 Equipment 9](#_Toc193793908)

[1.5 Staff 11](#_Toc193793909)

[1.6 Users 12](#_Toc193793910)

[1.7 Booking 14](#_Toc193793911)

[ЗАКЛЮЧЕНИЕ 15](#_Toc193793912)

# ВВЕДЕНИЕ

В данной лабораторной работе будет получен опыт по созданию SQL-запросов для выборки данных из реляционных баз данных, используя подзапросы, агрегатные функции, а также группировку данных с помощью оператора GROUP BY и операции над множествами (UNION, INTERSECT, MINUS).

Основная цель данной лабораторной работы – формирование практических навыков написания SQL-запросов, которые эффективно взаимодействуют с данными. При выполнении заданий важно учитывать правила написания запросов, включая ограничения на использование скалярных подзапросов и необходимость предварительного изучения данных в таблицах. Каждый запрос будет реализован в виде одного оператора SQL SELECT, который может включать подзапросы и группировку данных.

# SQL-ЗАПРОСЫ

## 1.1 Rehearsal\_points

Задание: *Вывести список* *репетиционных точек с рейтингом выше среднего.*

Скрипт для выведения списка репетиционных точек с рейтингом выше среднего:

SELECT name, rating FROM main.rehearsal\_points

WHERE rating > (

SELECT AVG(rating) FROM main.rehearsal\_points);

Таблица rehearsal\_points до скрипта представлена на рисунке 1.1.

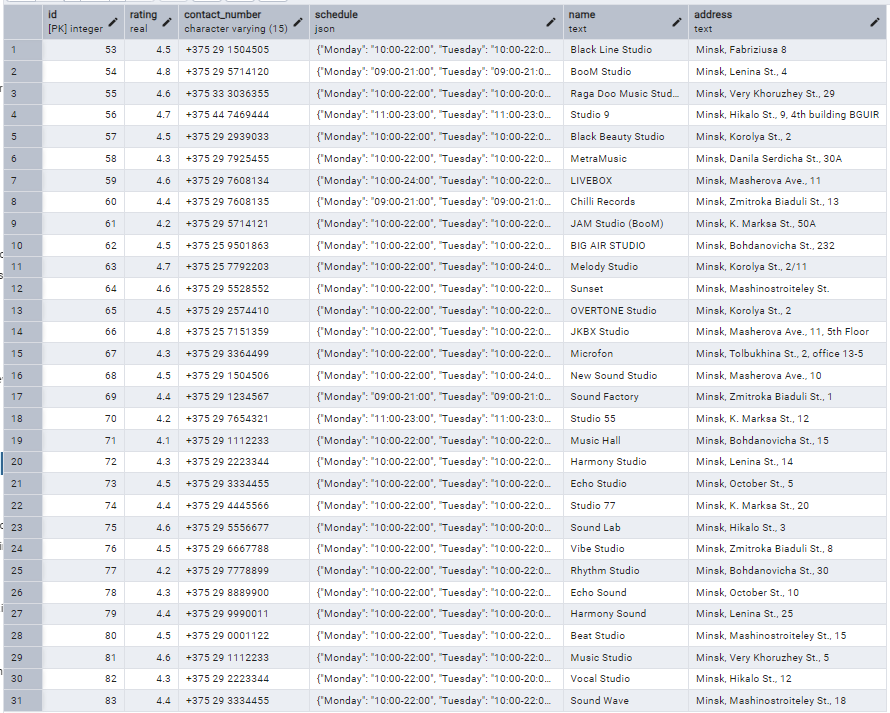


Рисунок 1.1 – Таблица rehearsal\_points до скрипта

Таблица rehearsal\_points после скрипта представлена на рисунке 1.2.

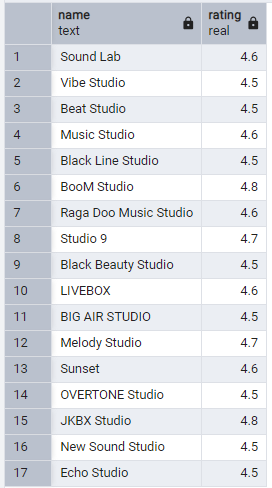
****

Рисунок 1.2 – Результат cкрипта

Задание: *Вывести* *названия репточек с комнатами, которые дешевле средней стоимости.*

Скрипт для выведения названий репточек с комнатами, которые дешевле средней стоимости:

SELECT rp.name, r.name, r.price

FROM main.rehearsal\_points rp

JOIN main.rooms r

ON r.id\_rehearsal\_point = rp.id

WHERE r.price <

(SELECT AVG(price) FROM main.rooms)

Таблица rehearsal\_points после скрипта представлена на рисунке 1.3.

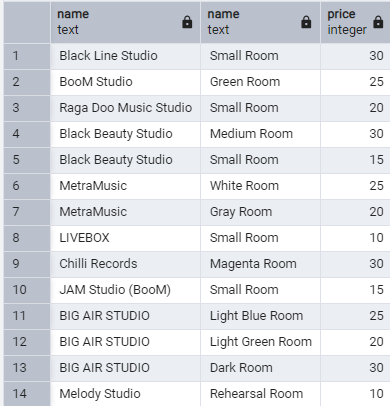
****

Рисунок 1.2 – Результат cкрипта

## 1.2 Rooms

Задание: *Вывести* *комнаты с ценой выше средней и возможностью звукозаписи.*

Скрипт для выведения комнаты с ценой выше средней и возможностью звукозаписи:

SELECT id, name, price

FROM main.rooms

WHERE price >

(SELECT AVG(price) FROM main.rooms)

AND recording\_support = true;

Таблица rooms до скрипта представлена на рисунке 1.4.

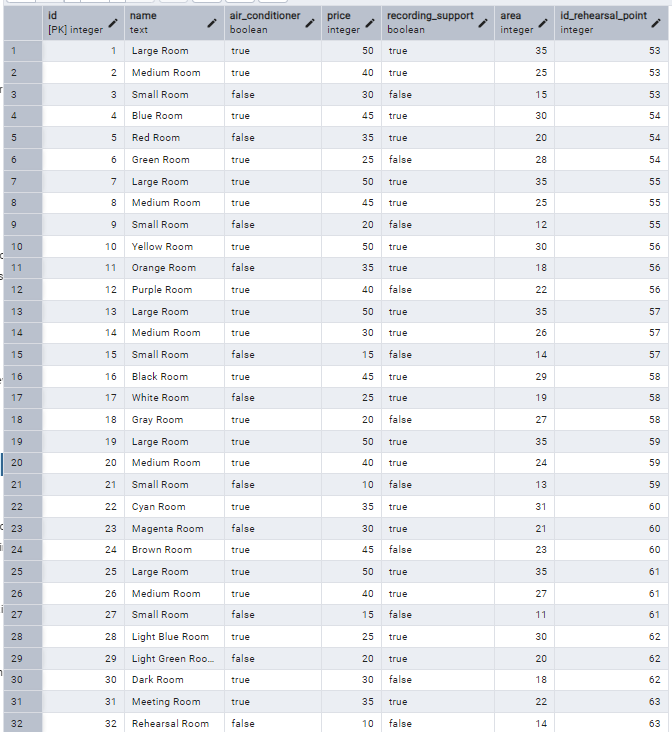


Рисунок 1.4 – Таблица rooms до скрипта

Таблица rooms после скрипта представлена на рисунке 1.5.

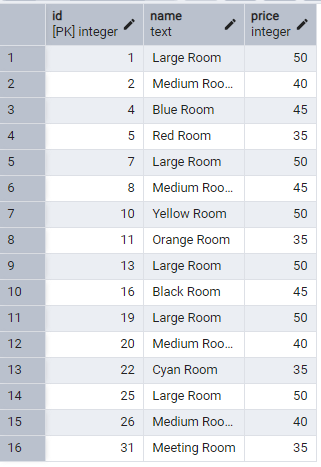
****

Рисунок 1.5 – Результат cкрипта

Задание: *Вывести количество комнат с кондиционером в репточках, где такие комнаты есть.*

Скрипт для выведения количества комнат с кондиционером в репточках, где такие комнаты есть:

SELECT rp.name, COUNT(\*) AS air\_conditioned\_rooms

FROM main.rooms r

JOIN main.rehearsal\_points rp

ON r.id\_rehearsal\_point = rp.id

WHERE air\_conditioner = TRUE

GROUP BY rp.name;

Таблица rooms после скрипта представлена на рисунке 1.6.

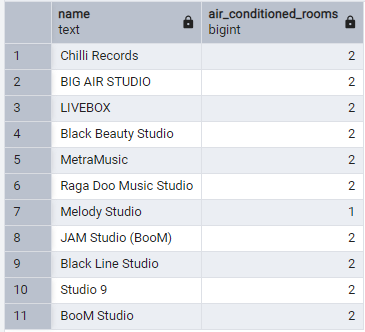
****

Рисунок 1.6 – Результат cкрипта

## 1.3 Service

Задание: *Вывести количество услуг репточек и список различных типов услуг.*

Скрипт для выведения количества услуг репточек и список различных типов услуг:

SELECT rp.name,

COUNT(\*) AS service\_count,

STRING\_AGG(DISTINCT s.type, ', ')

FROM main.service s

JOIN main.rehearsal\_points rp

ON s.id\_rehearsal\_point = rp.id

GROUP BY rp.name;

Таблица service до скрипта представлена на рисунке 1.7.



Рисунок 1.7 – Таблица service до скрипта

Таблица service после скрипта представлена на рисунке 1.8.

****

Рисунок 1.8 – Результат cкрипта

Задание: *Вывести список* *услуг, чья цена выше средней для своего типа.*

Скрипт для выведения списка услуг, чья цена выше средней для своего типа:

SELECT id, name, price

FROM main.service s

WHERE price >

(SELECT AVG(price)

FROM main.service sub\_s

WHERE s.type = sub\_s.type);

Таблица service после скрипта представлена на рисунке 1.9.

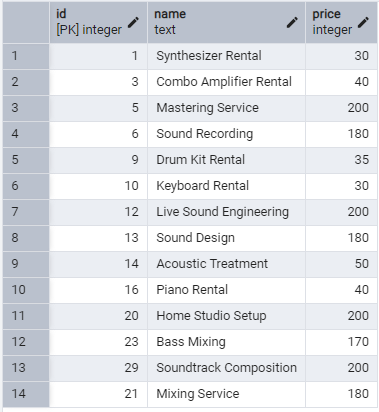


Рисунок 1.9 – Результат cкрипта

## 1.4 Equipment

Задание: *Вывести количество оборудования по репточкам.*

Скрипт для выведения количества оборудования по репточкам:

SELECT rp.name, COUNT(\*) AS equipment\_count

FROM main.equipment e

JOIN main.rehearsal\_points rp ON rp.id = e.id\_rehearsal\_point

GROUP BY rp.name;

Таблица equipment до скрипта представлена на рисунке 1.10.

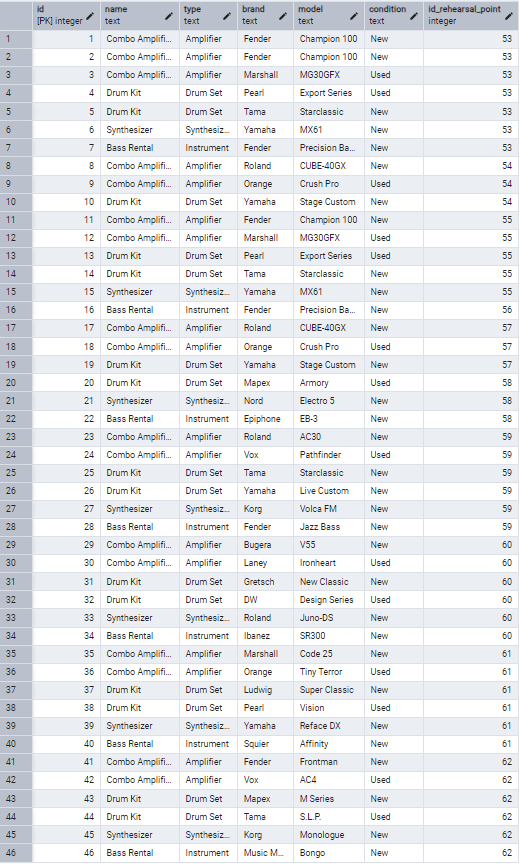


Рисунок 1.10 – Таблица equipment до скрипта

Таблица equipment после скрипта представлена на рисунке 1.11.

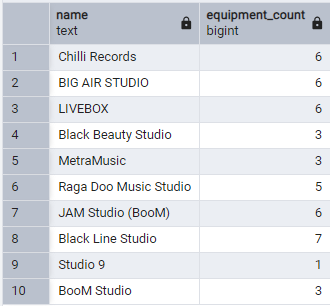
****

Рисунок 1.11 – Результат cкрипта

Задание: *Вывести список оборудования, которое не указано в активных бронированиях.*

Скрипт для выведения списка оборудования, которое не указано в активных бронированиях.

SELECT name, id

FROM main.equipment

WHERE id NOT IN (

SELECT id\_equipment

FROM main.equipment\_booking);

Таблица equipment после скрипта представлена на рисунке 1.12.



Рисунок 1.12 – Результат cкрипта

## 1.5 Staff

Задание: *Вывести средний возраст сотрудников по репточкам.*

Скрипт для выведения среднего возраста сотрудников по репточкам:

SELECT id\_rehearsal\_point, ROUND(AVG(age), 2) AS average\_age

FROM main.staff

GROUP BY id\_rehearsal\_point;

Таблица staff до скрипта представлена на рисунке 1.13.

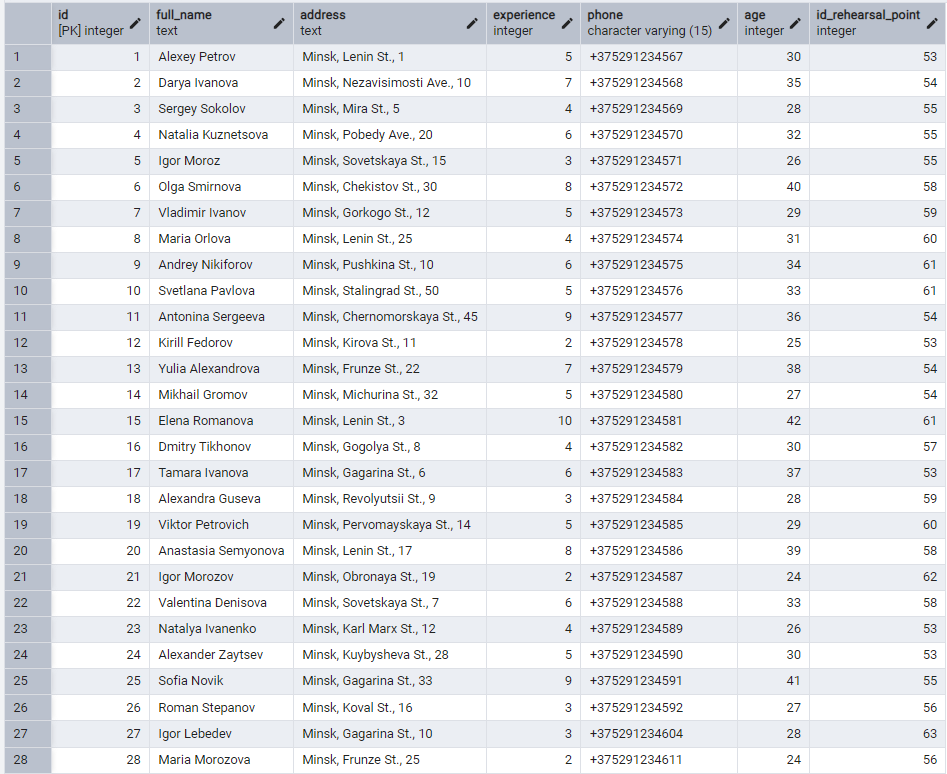


Рисунок 1.13 – Таблица staff до скрипта

Таблица staff после скрипта представлена на рисунке 1.14.

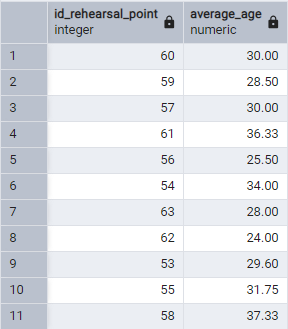
****

Рисунок 1.14 – Результат cкрипта

Задание: *Вывести количество сотрудников по возрастным категориям.*

Скрипт для выведения количества сотрудников по возрастным категориям:

WITH age\_groups AS (

SELECT 'Under 20' AS age\_group, 0 AS lower\_bound, 19 AS upper\_bound

UNION ALL SELECT '20-25', 20, 25

UNION ALL SELECT '25-30', 26, 30

UNION ALL SELECT '30-35', 31, 35

UNION ALL SELECT '35-40', 36, 40

UNION ALL SELECT 'Above 40', 41, 100

)

SELECT ag.age\_group, COUNT(s.age) AS staff\_count

FROM age\_groups ag

LEFT JOIN main.staff s ON s.age BETWEEN ag.lower\_bound AND ag.upper\_bound

GROUP BY ag.age\_group

ORDER BY staff\_count DESC;

Таблица staff после скрипта представлена на рисунке 1.15.

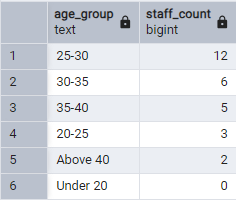


Рисунок 1.15 – Результат cкрипта

## 1.6 Users

Задание: *Вывести* *количество зарегистрировавшихся пользователей по месяцам.*

Скрипт для выведения количества зарегистрировавшихся пользователей по месяцам:

SELECT DATE\_TRUNC('month', registration\_date) AS registration\_month, COUNT(\*) AS user\_count

FROM main.users

GROUP BY registration\_month

ORDER BY registration\_month;

Таблица users до скрипта представлена на рисунке 1.16.



Рисунок 1.16 – Таблица users до скрипта

Таблица users после скрипта представлена на рисунке 1.17.

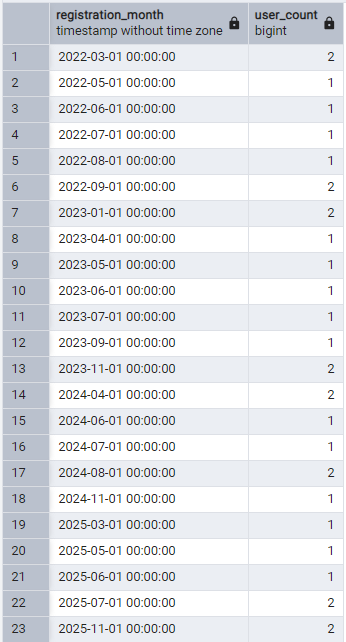
****

Рисунок 1.17 – Результат cкрипта

## 1.7 Booking

Задание: *Вывести* *количество бронирований по статусу.*

Скрипт для выведения количества бронирований по статусу:

SELECT status, COUNT(\*) AS booking\_count

FROM main.booking

GROUP BY status;

Таблица booking до скрипта представлена на рисунке 1.18.



Рисунок 1.18 – Таблица booking до скрипта

Таблица booking после скрипта представлена на рисунке 1.19.

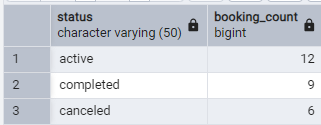
****

Рисунок 1.19 – Результат cкрипта

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была успешно реализована серия SQL-запросов для анализа данных из базы данных, связанной с репетиционными базами. Работа включала создание подзапросов, применение агрегатных функций и группировку данных, что значительно углубило как теоретические знания, так и практические навыки работы с реляционными базами данных.

Результаты лабораторной работы продемонстрировали эффективность применения сложных SQL-запросов для получения аналитической информации, а также важность методов группировки и агрегирования для формирования отчетов. Использование подзапросов позволило более точно фильтровать необходимую информацию и лучше понять структуру данных и их взаимосвязи в базе данных.

Эта лабораторная работа подчеркнула значимость правильного проектирования баз данных и выбора типов данных для обеспечения целостности и эффективности хранения информации. Приобретенные навыки окажутся полезными для решения реальных задач анализа данных, включая создание отчетов и статистики. В дальнейшем рекомендуется сосредоточиться на изучении индексов для оптимизации запросов, а также освоении более сложных аспектов языка SQL, таких как триггеры и процедуры. Таким образом, работа стала важной основой для дальнейшего изучения и практики в области управления базами данных и анализа данных.

.