Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Тема «Грузоперевозки»

Лабораторная работа №5

Реализация SQL-запросов на выборку данных с группировкой результатов

Студент: Е.О. Лукьянов

Преподаватель: Д.В. Куприянова

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc194583423)

[1 SQL-ЗАПРОСЫ 4](#_Toc194583424)

[1.1 Транспортные средства 4](#_Toc194583425)

[1.2 Транспортные операции 6](#_Toc194583426)

[1.3 Грузы 8](#_Toc194583427)

[1.4 Клиенты 10](#_Toc194583428)

[1.5 Водители 12](#_Toc194583429)

[1.6 Оплаты 13](#_Toc194583430)

[1.7 Маршруты 14](#_Toc194583431)

[ЗАКЛЮЧЕНИЕ 19](#_Toc194583432)

# ВВЕДЕНИЕ

В данной лабораторной работе будет получен опыт по создание SQL-запросы 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). Работа включает несколько этапов, начиная с получения заданий от преподавателя на основе схемы данных, созданной в лабораторной работе №2, и реализованной в виде таблиц в СУБД в лабораторной работе №3.

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.

Эта лабораторная работа поможет углубить понимание работы с SQL и подготовит нас к более сложным задачам анализа и обработки данных в реляционных базах данных.

# 1 SQL-ЗАПРОСЫ

## 1.1 Транспортные средства

Задание: *Вывести модель и суммарную грузоподъёмность всех транспортных средств по каждой модели по возрастанию.*

Скрипт для вывода модели и суммарной грузоподъёмности всех транспортных средств по каждой модели по возрастанию.

SELECT "Модель", SUM("Грузоподъёмность") AS Суммарная\_грузоподъемность

FROM "Транспортные средства"

GROUP BY "Модель"

ORDER BY Суммарная\_грузоподъемность ASC;

Таблица «Транспортные средства» до скрипта представлена на рисунке 1.1.

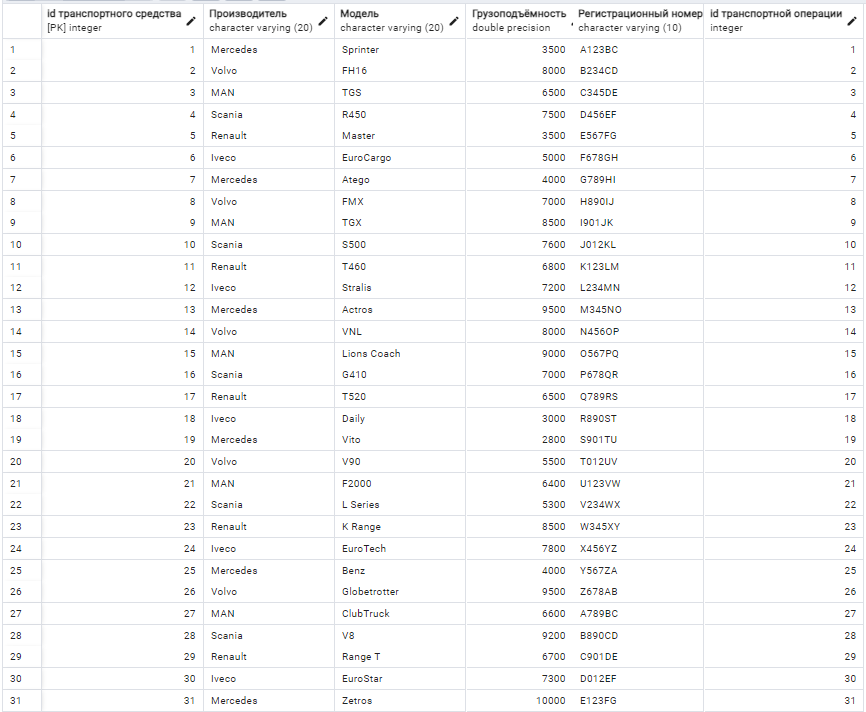


Рисунок 1.1 – Таблица «Транспортные средства» до скрипта

Таблица «Транспортные средства» после скрипта представлена на рисунке 1.2.



Рисунок 1.2 – Результат скрипта

Задание: *Вывести производителя и максимальную грузоподъёмность среди его моделей по убыванию.*

Скрипт для вывода производителей и максимальной грузоподъёмности среди их моделей по убыванию.

SELECT "Производитель", MAX("Грузоподъёмность") AS Максимальная\_грузоподъемность

FROM "Транспортные средства"

GROUP BY "Производитель"

ORDER BY Максимальная\_грузоподъемность DESC;

Таблица «Транспортные средства» до скрипта представлена на рисунке 1.1.

Таблица «Транспортные средства» после скрипта представлена на рисунке 1.3.

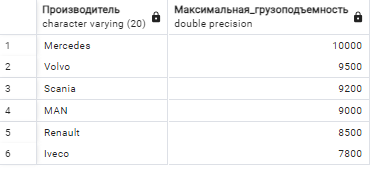


Рисунок 1.3 – Результат скрипта

## 1.2 Транспортные операции

Задание: *Вывести суммарную стоимость всех оплат по каждому статусу транспортной операции.*

Скрипт для вывода суммарной стоимости всех оплат по каждому статусу транспортной операции.

SELECT то."Статус", SUM(o."Сумма") AS Суммарная\_стоимость

FROM "Транспортные операции" то

JOIN "Оплаты" o ON то."id транспортной операции" = o."id транспортной операции"

GROUP BY то."Статус";

Таблица «Транспортные операции» до скрипта представлена на рисунке 1.4.

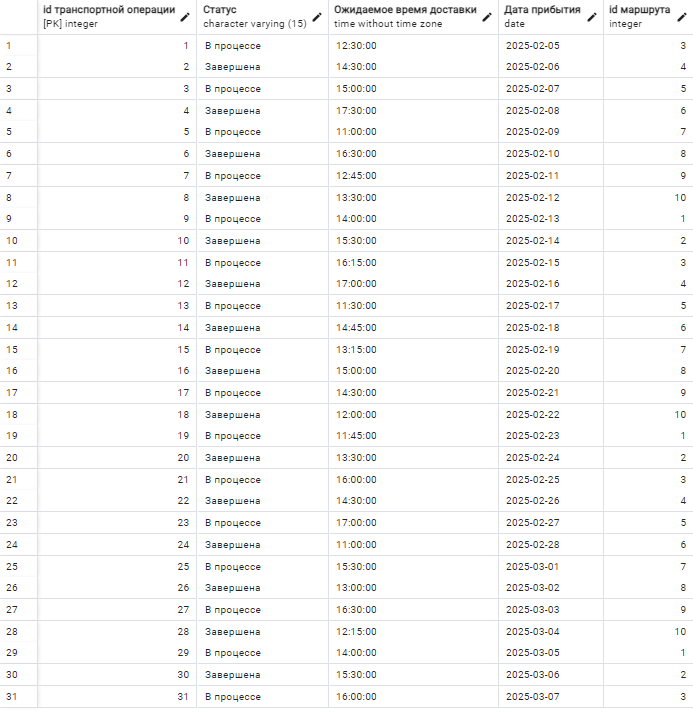


Рисунок 1.4 – Таблица «Транспортные операции» до скрипта

Таблица «Транспортные операции» после скрипта представлена на рисунке 1.5.



Рисунок 1.5 – Результат скрипта

Задание: *Вывести среднюю стоимость оплаты для каждого маршрута по возрастанию.*

Скрипт для вывода средней стоимости оплаты для каждого маршрута по возрастанию.

SELECT m."Точка отправления", m."Точка прибытия",

ROUND(AVG(o."Сумма")) AS Средняя\_стоимость

FROM "Маршруты" m

JOIN "Транспортные операции" то ON m."id маршрута" = то."id маршрута"

JOIN "Оплаты" o ON то."id транспортной операции" = o."id транспортной операции"

GROUP BY m."Точка отправления", m."Точка прибытия"

ORDER BY Средняя\_стоимость ASC;

Таблица «Транспортные операции» до скрипта представлена на рисунке 1.4.

Таблица «Транспортные операции» после скрипта представлена на рисунке 1.6.

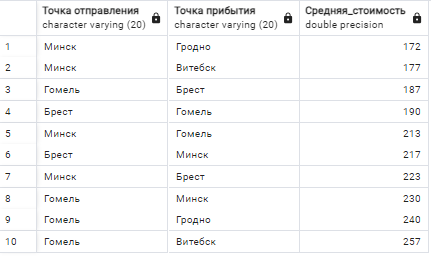


Рисунок 1.6 – Результат скрипта

## 1.3 Грузы

Задание: *Вывести средний вес грузов по каждой точке отправления.*

Скрипт для вывода среднего веса грузов по каждой точке отправления*.*

SELECT m."Точка отправления", ROUND(AVG(g."Вес")) AS Средний\_вес

FROM "Грузы" g

JOIN "Транспортные операции" то ON g."id транспортной операции" = то."id транспортной операции"

JOIN "Маршруты" m ON то."id маршрута" = m."id маршрута"

GROUP BY m."Точка отправления";

Таблица «Грузы» до скрипта представлена на рисунке 1.7.

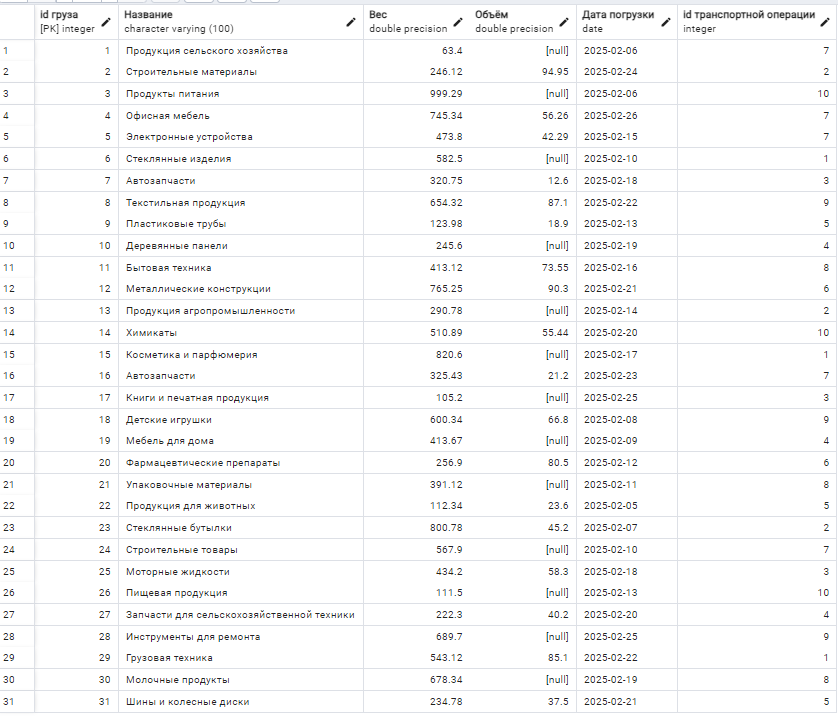


Рисунок 1.7 – Таблица «Грузы» до скрипта

Таблица «Грузы» после скрипта представлена на рисунке 1.8.

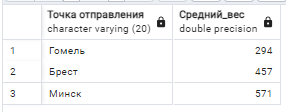


Рисунок 1.8 – Результат скрипта

Задание: *Вывести количество грузов с объёмом более 50м3 по каждой точке прибытия, при условии что их количество больше 1.*

Скрипт для вывода количества грузов с объёмом более 50м3 по каждой точке прибытия, при условии что их количество больше 1*.*

SELECT m."Точка прибытия", COUNT(g."id груза") AS Количество\_грузов

FROM "Грузы" g

JOIN "Транспортные операции" то ON g."id транспортной операции" = то."id транспортной операции"

JOIN "Маршруты" m ON то."id маршрута" = m."id маршрута"

WHERE g."Объём" > 50

GROUP BY m."Точка прибытия"

HAVING COUNT(g."id груза") > 1;

Таблица «Грузы» до скрипта представлена на рисунке 1.7.

Таблица «Грузы» после скрипта представлена на рисунке 1.9.

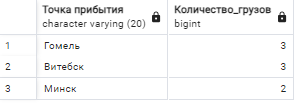


Рисунок 1.9 – Результат скрипта

## 1.4 Клиенты

Задание: *Вывести количество транспортных операций для клиентов из Минска и Гомеля.*

Скрипт для вывода количества транспортных операций для клиентов из Минска и Гомеля.

SELECT

CASE

WHEN k."Адрес" LIKE '%Минск%' THEN 'Минск'

WHEN k."Адрес" LIKE '%Гомель%' THEN 'Гомель'

END AS Город,

COUNT(kto."id транспортной операции") AS Количество\_операций

FROM "Клиенты" k

JOIN "Клиенты-Транспортные операции" kto ON k."id клиента" = kto."id клиента"

WHERE k."Адрес" LIKE '%Минск%' OR k."Адрес" LIKE '%Гомель%'

GROUP BY CASE

WHEN k."Адрес" LIKE '%Минск%' THEN 'Минск'

WHEN k."Адрес" LIKE '%Гомель%' THEN 'Гомель'

END;

Таблица «Клиенты» до скрипта представлена на рисунке 1.10.

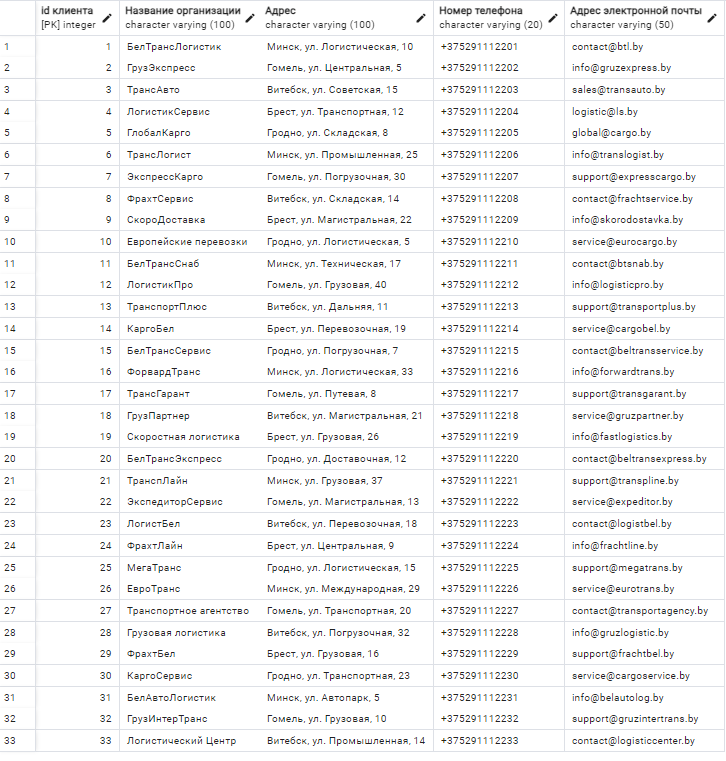


Рисунок 1.10 – Таблица «Клиенты» до скрипта

Таблица «Клиенты» после скрипта представлена на рисунке 1.11.

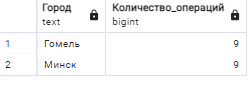


Рисунок 1.11 – Результат скрипта

## 1.5 Водители

Задание: *Вывести возрастные категории водителей и количество водителей в каждой категории, где количество больше 10, отсортировать по убыванию.*

Скрипт для вывода возрастных категорий водителей и количество водителей в каждой категории, где количество больше 10, отсортировать по убыванию.

SELECT

CASE

WHEN "Возраст" < 30 THEN 'До 30 лет'

WHEN "Возраст" BETWEEN 30 AND 40 THEN '30-40 лет'

WHEN "Возраст" > 40 THEN 'Старше 40 лет'

END AS Возрастная\_категория,

COUNT(\*) AS Количество\_водителей

FROM "Водители"

GROUP BY CASE

WHEN "Возраст" < 30 THEN 'До 30 лет'

WHEN "Возраст" BETWEEN 30 AND 40 THEN '30-40 лет'

WHEN "Возраст" > 40 THEN 'Старше 40 лет'

END

HAVING COUNT(\*) > 10

ORDER BY Количество\_водителей DESC;

Таблица «Водители» до скрипта представлена на рисунке 1.12.

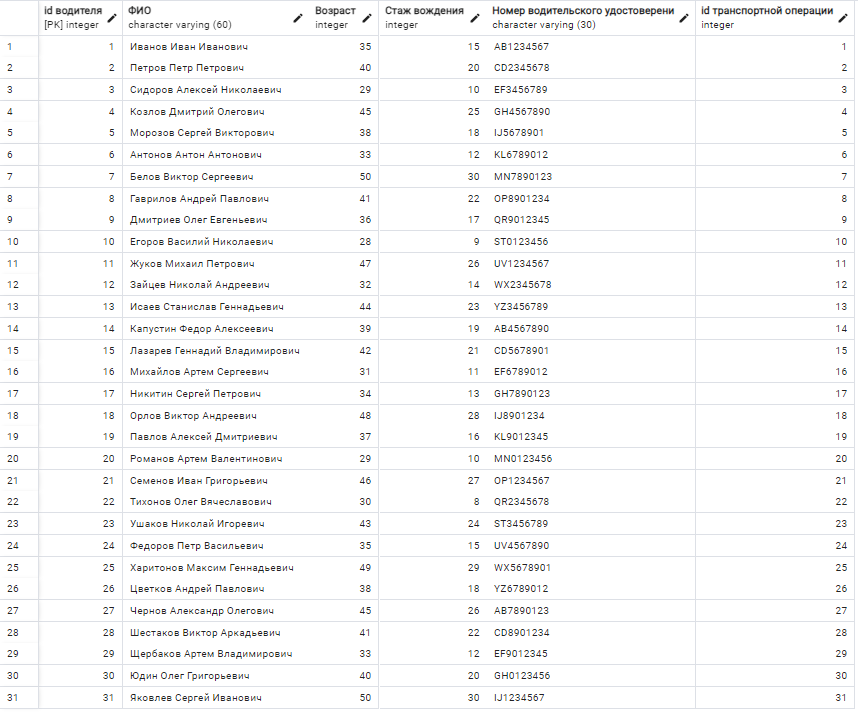


Рисунок 1.12 – Таблица «Водители» до скрипта

Таблица «Водители» после скрипта представлена на рисунке 1.13.

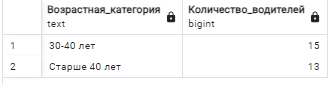


Рисунок 1.13 – Результат скрипта

## 1.6 Оплаты

Задание: *Вывести максимальную сумму оплаты по каждому способу оплаты.*

Скрипт для вывода максимальной суммы оплаты по каждому способу оплаты.

SELECT MAX(Стоимость) AS максимальная\_стоимость, Способ

FROM Оплата

GROUP BY Способ

Таблица «Оплаты» до скрипта представлена на рисунке 1.14.

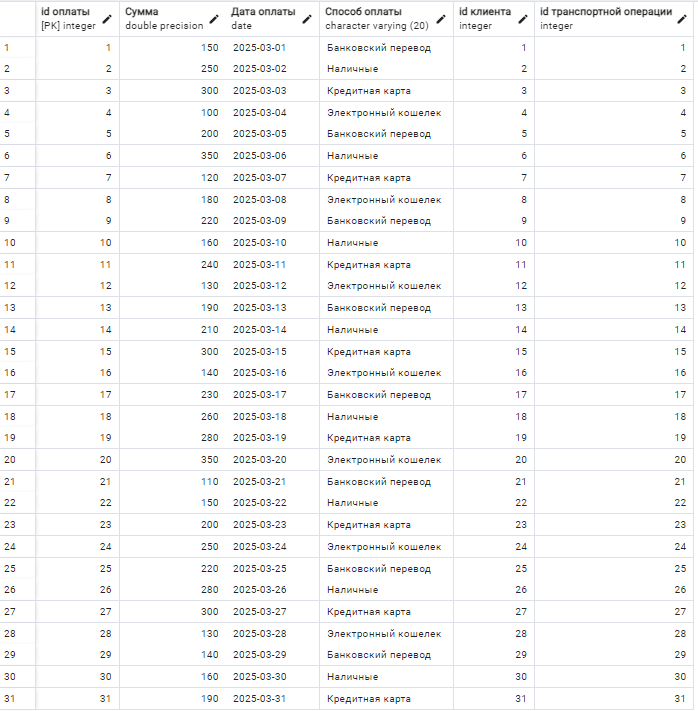


Рисунок 1.14 – Таблица «Оплаты» до скрипта

Таблица «Оплаты» после скрипта представлена на рисунке 1.15.

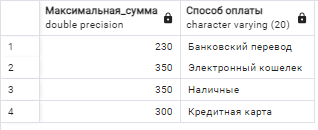


Рисунок 1.15– Результат скрипта

Задание: *Вывести сумму оплат, которые будут произведены после 2025-03-17, по каждому способу оплаты.*

Скрипт для вывода суммы оплат, которые будут произведены после 2025-03-17, по каждому способу оплаты.

SELECT SUM("Сумма") AS Суммарная\_стоимость, "Способ оплаты"

FROM "Оплаты"

WHERE "Дата оплаты" > '2025-03-17'

GROUP BY "Способ оплаты";

Таблица «Оплаты» до скрипта представлена на рисунке 1.14.

Таблица «Оплаты» после скрипта представлена на рисунке 1.16.

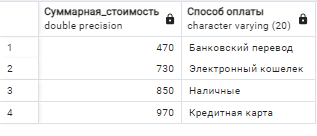


Рисунок 1.16– Результат скрипта

## 1.7 Маршруты

Задание: *Вывести среднее расстояние маршрутов между каждой парой городов, округленное до целого числа, отсортированные по убыванию.*

Скрипт для вывода среднего расстояния маршрутов между каждой парой городов, округленного до целого числа, отсортированные по убыванию.

SELECT

"Точка отправления",

"Точка прибытия",

ROUND(AVG("Расстояние")) AS Среднее\_расстояние

FROM "Маршруты"

GROUP BY "Точка отправления", "Точка прибытия"

ORDER BY Среднее\_расстояние DESC;

Таблица «Маршруты» до скрипта представлена на рисунке 1.17.



Рисунок 1.17– Таблица «Маршруты» до скрипта

Таблица «Маршруты» после скрипта представлена на рисунке 1.18.

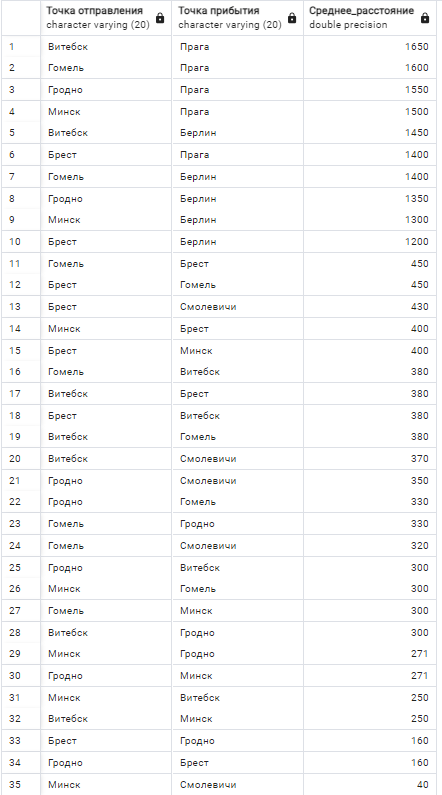


Рисунок 1.18 – Результат скрипта

Задание: *Вывести количество маршрутов для каждой точки отправления, где время в пути превышает 5 часов, отсортировать по возрастанию количества.*

Скрипт для вывода количества маршрутов для каждой точки отправления, где время в пути превышает 5 часов, отсортированные по возрастанию количества.

SELECT

"Точка отправления",

COUNT(\*) AS Количество\_маршрутов

FROM "Маршруты"

WHERE "Время в пути" > '05:00:00'

GROUP BY "Точка отправления"

ORDER BY Количество\_маршрутов ASC;

Таблица «Маршруты» до скрипта представлена на рисунке 1.17.

Таблица «Маршруты» после скрипта представлена на рисунке 1.19.

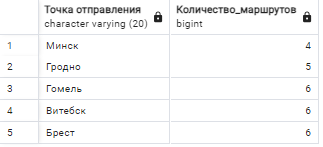


Рисунок 1.19 – Результат скрипта

Задание: *Вывести 3 самых длинных маршрута по расстоянию для каждой точки отправления, с указанием точки прибытия и расстояния.*

Скрипт для вывода 3 самых длинных маршрутов по расстоянию для каждой точки отправления, с указанием точки прибытия и расстояния.

SELECT \*

FROM (

SELECT

"Точка отправления",

"Точка прибытия",

"Расстояние",

ROW\_NUMBER() OVER (PARTITION BY "Точка отправления" ORDER BY "Расстояние" DESC) AS rank

FROM "Маршруты"

) ranked\_routes

WHERE rank <= 3;

Таблица «Маршруты» до скрипта представлена на рисунке 1.17.

Таблица «Маршруты» после скрипта представлена на рисунке 1.20.



Рисунок 1.20 – Результат скрипта

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были успешно реализованы SQL-запросы для выборки данных из реляционной базы данных туристического агентства. Основной целью работы стало освоение базовых операций языка SQL, включая фильтрацию, сортировку, соединение таблиц и использование скалярных функций.

Ключевые результаты:

* + 1. Фильтрация и сортировка данных.
    2. Соединение таблиц (JOIN).
    3. Работа с датами.
    4. Валидация данных.

Работа подтвердила, что грамотное проектирование структуры базы данных и использование индексов значительно упрощают выполнение сложных запросов. Освоенные навыки позволяют эффективно извлекать и анализировать данные, что критически важно для автоматизации процессов в туристическом бизнесе (сбор информации о клиентах и сотрудниках, управление платежами, планирование ресурсов).

Выполненная работа стала важным этапом в освоении технологий работы с базами данных и заложила основу для изучения более сложных SQL-операций, таких как группировка, агрегация и оптимизация запросов.