Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе № 5

РЕАЛИЗАЦИЯ SQL-ЗАПРОСОВ НА ВЫБОРКУ С ГРУППИРОВКОЙ РЕЗУЛЬТАТОВ

на тему

«Сеть клубов карате»

Студент: Р.С. Кочеров

Преподаватель: Д.В. Куприянова

МИНСК 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc193973166)

[1. ВЫПОЛНЕНИЕ ЗАПРОСОВ 4](#_Toc193973167)

[1.1 Запрос 1 4](#_Toc193973168)

[1.2 Запрос 2 5](#_Toc193973169)

[1.3 Запрос 3 6](#_Toc193973170)

[1.4 Запрос 4 8](#_Toc193973171)

[1.5 Запрос 5 10](#_Toc193973172)

[1.6 Запрос 6 11](#_Toc193973173)

[1.7 Запрос 7 13](#_Toc193973174)

[1.8 Запрос 8 14](#_Toc193973175)

[1.9 Запрос 9 15](#_Toc193973176)

[1.10 Запрос 10 17](#_Toc193973177)

[1.11 Запрос 11 18](#_Toc193973178)

[1.12 Запрос 12 20](#_Toc193973179)

[1.13 Запрос 13 21](#_Toc193973180)

[1.14 Запрос 14 22](#_Toc193973181)

[ЗАКЛЮЧЕНИЕ 24](#_Toc193973182)

# **ВВЕДЕНИЕ**

Для работы с данными используются системы управления базами данных (СУБД), которые позволяют хранить, обрабатывать и извлекать информацию эффективно и структурированно. Одним из основных инструментов взаимодействия с базами данных является язык структурированных запросов. SQL предоставляет широкие возможности для выборки, фильтрации, сортировки и агрегации данных, что делает его незаменимым инструментом администраторов баз данных.

Цель данной лабораторной работы — изучение и освоение базовых и продвинутых принципов составления SQL-запросов с использованием таких элементов, как подзапросы, агрегатные функции, группировка данных с помощью оператора GROUP BY, а также операций над множествами (UNION, INTERSECT, MINUS). Рассматриваемые инструменты позволяют решать сложные задачи, связанные с анализом данных, вычислениями и их представлением в удобной форме.

Во время выполнения лабораторной работы используются заранее подготовленные таблицы, моделирующие определённую предметную область. Эти таблицы содержат разнообразные данные, которые будут обработаны с помощью SQL-запросов. На основе данных из таблиц будут составлены запросы, демонстрирующие возможности извлечения, группировки, фильтрации и объединения данных. Также будут применены подзапросы для создания многоуровневых запросов и выполнения дополнительных вычислений.

Освоение материала лабораторной работы позволит не только углубить теоретические знания, но и получить практические навыки работы с SQL. Это важный этап в изучении технологий работы с реляционными базами данных, который закладывает основу для выполнения более сложных аналитических задач, проектирования баз данных и их оптимизации. Имея представление о подзапросах, агрегатных функциях и операциях над множествами, можно эффективно решать широкий спектр задач, связанных с обработкой данных.

# **1. ВЫПОЛНЕНИЕ ЗАПРОСОВ**

## **1.1 Запрос 1**

Группировка по стилю карате и подсчет количества клубов с таким стилем.

SELECT karate\_style, COUNT(id) as club\_count

FROM club

GROUP BY karate\_style

ORDER BY club\_count DESC;

Изображение выглядит как текст, меню, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1.1 – Таблица club

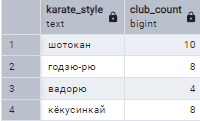


Рисунок 1.2 – Результат выполнения запроса

## **1.2 Запрос 2**

Группировка по мастерству тренеров и подсчет суммы их заработных плат.

SELECT mastery\_level\_coach, SUM(salary) AS total\_salary

FROM public.coach

GROUP BY mastery\_level\_coach;

Изображение выглядит как текст, меню, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1.3 – Таблица coach

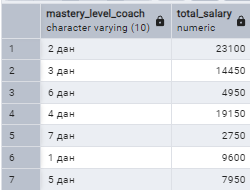


Рисунок 1.4 – Результат выполнения запроса

## **1.3 Запрос 3**

Группировка по типу тренировок и расчет количества учеников, которые их посещают.

SELECT w.type, COUNT(sw.student\_key) AS student\_count

FROM public.workout w

JOIN public.student\_workout sw ON w.id = sw.workout\_key

GROUP BY w.type;

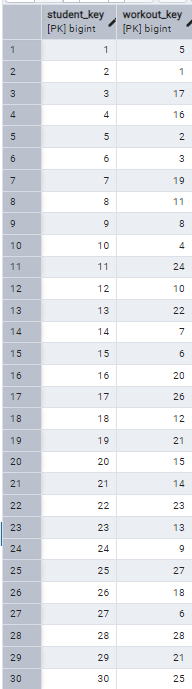


Рисунок 1.5 – Таблица workout

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

1.6 – Таблица student



1.7 – Таблица student\_workout

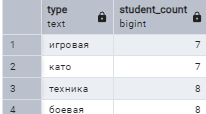


Рисунок 1.8 – Результат выполнения запроса

## **1.4 Запрос 4**

Группировка по типу мероприятия и расчет количества учеников, которые их посещают (возраст < 20).

SELECT e.type\_event, COUNT(se.student\_key) AS student\_count

FROM public.event e

JOIN public.student\_event se ON e.id = se.event\_key

JOIN public.student s ON se.student\_key = s.id

WHERE s.age < 20

GROUP BY e.type\_event;

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

1.9 – Таблица student

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1.10 – Таблица student\_event

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

1.11 – Таблица event

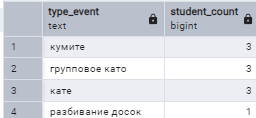


Рисунок 1.12 – Результат выполнения запроса

## **1.5 Запрос 5**

Выборка по типу оплаты и расчет количества учеников, которые так платят.

SELECT type\_payment, COUNT(DISTINCT student\_id\_payment) AS student\_count

FROM public.payment

GROUP BY type\_payment;

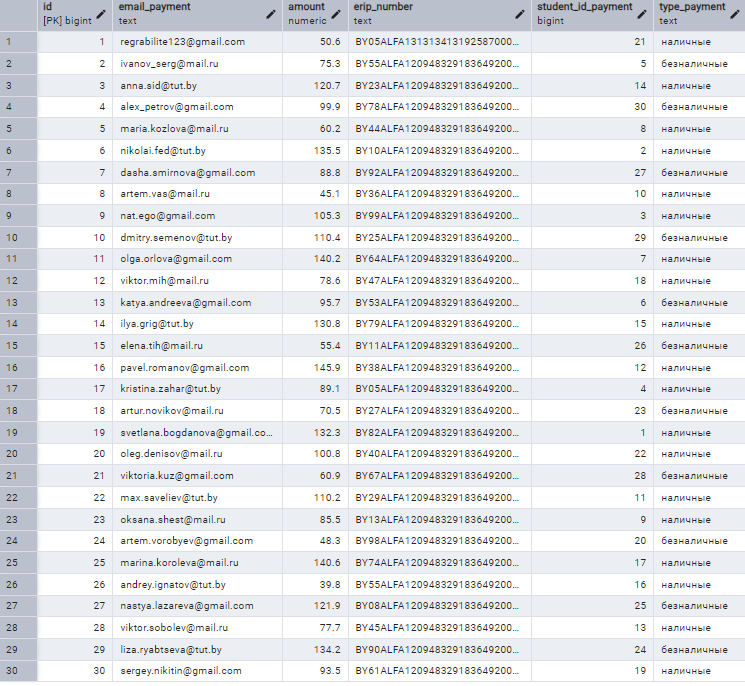


Рисунок 1.13 – Таблица payment

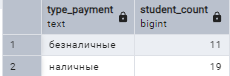


Рисунок 1.14 – Результат выполнения запроса

## **1.6 Запрос 6**

Выбор клубов, участвующих в мероприятиях уровня «международный».

SELECT c.name\_club

FROM public.club c

JOIN public.event e ON c.id = e.club\_id\_event

WHERE e.level = 'международный'

GROUP BY c.name\_club;

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

1.15 – Таблица event

Изображение выглядит как текст, меню, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1.16 – Таблица club



Рисунок 1.17 – Результат выполнения запроса

## **1.7 Запрос 7**

Выбор мест проведения мероприятия, у которых вместимость больше средней.

SELECT el.id, el.address, el.capacity

FROM public.event\_location el

GROUP BY el.id, el.address, el.capacity

HAVING el.capacity > (SELECT AVG(capacity) FROM public.event\_location);

Изображение выглядит как текст, меню, снимок экрана, документ

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1.18 – Таблица event\_location

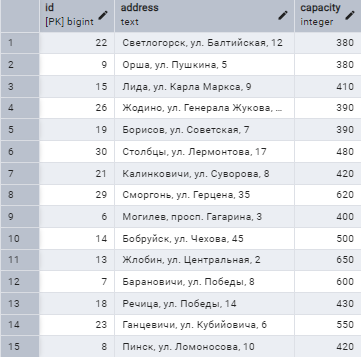


Рисунок 1.19 – Результат выполнения запроса

## **1.8 Запрос 8**

Группировка по мастерству тренеров и сортировка по возрастанию.

SELECT mastery\_level\_coach, COUNT(\*) AS coach\_count

FROM public.coach

GROUP BY mastery\_level\_coach

ORDER BY mastery\_level\_coach ASC;



Рисунок 1.20 – Таблица coach

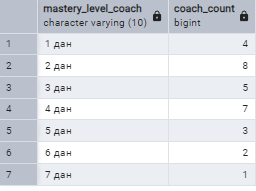


Рисунок 1.21 – Результат выполнения запроса 8

## **1.9 Запрос 9**

Группировка по возрасту учеников и подсчет количества мероприятий, в которых они приняли участие.

SELECT s.age, COUNT(se.event\_key) AS event\_count

FROM public.student s

JOIN public.student\_event se ON s.id = se.student\_key

GROUP BY s.age;

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

1.22 – Таблица student

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1.23 – Таблица student\_event

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

1.24 – Таблица event

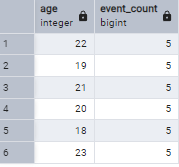


Рисунок 1.25 – Результат выполнения

## **1.10 Запрос 10**

Вывести мероприятия, у которых цена аренды места проведения больше 600.

SELECT e.id, e.type\_event, e.date\_event, e.level

FROM public.event e

JOIN public.event\_location el ON e.evenlocation\_id\_event = el.id

GROUP BY e.id, e.type\_event, e.date\_event, e.level, el.price

HAVING el.price > 600;

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

1.26 – Таблица event

Изображение выглядит как текст, меню, снимок экрана, документ

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1.27 – Таблица event\_location

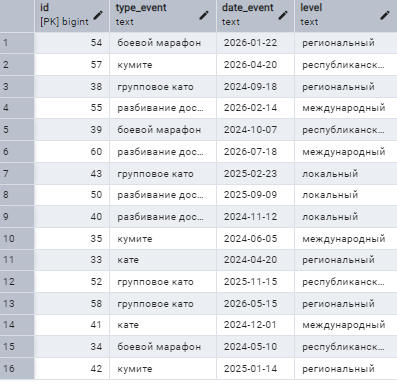


Рисунок 1.28 – Результат выполнения запроса

## **1.11 Запрос 11**

Группировка по группе тренировок с указанием количества тренеров, которые проводят эти тренировки.

SELECT w.group\_name, COUNT(DISTINCT w.coach\_id\_workout) AS coach\_count

FROM public.workout w

GROUP BY w.group\_name;



Рисунок 1.29 – Таблица coach



Рисунок 1.30 – Таблица workout

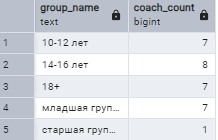


Рисунок 1.31 – Результат выполнения запроса

## **1.12 Запрос 12**

Группировка по уровню мастерства учеников и подсчет их количества.

SELECT mastery\_level\_student, COUNT(\*) AS student\_count

FROM public.student

GROUP BY mastery\_level\_student;

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

1.32 – Таблица student

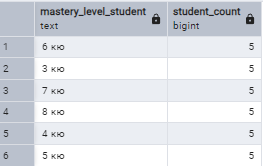


Рисунок 1.33 – Результат выполнения запроса

## **1.13 Запрос 13**

Выборка по спонсорам мероприятий и подсчет количества мероприятий, которые они спонсировали.

SELECT sponsor, COUNT(\*) AS event\_count

FROM public.event

GROUP BY sponsor;

Изображение выглядит как текст, снимок экрана, меню, число

Контент, сгенерированный ИИ, может содержать ошибки.

1.34 – Таблица event

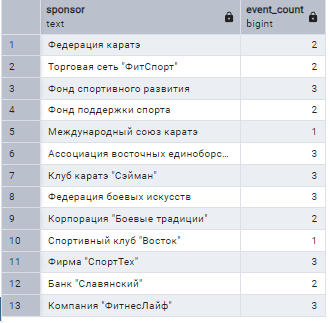


Рисунок 1.35 – Результат выполнения запроса

## **1.14 Запрос 14**

Группировка по количеству учеников у тренера.

SELECT c.name\_coach, COUNT(s.id) AS student\_count

FROM public.coach c

LEFT JOIN public.student s ON c.id = s.coach\_id\_student

GROUP BY c.name\_coach;

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

1.36 – Таблица student



Рисунок 1.37 – Таблица coach

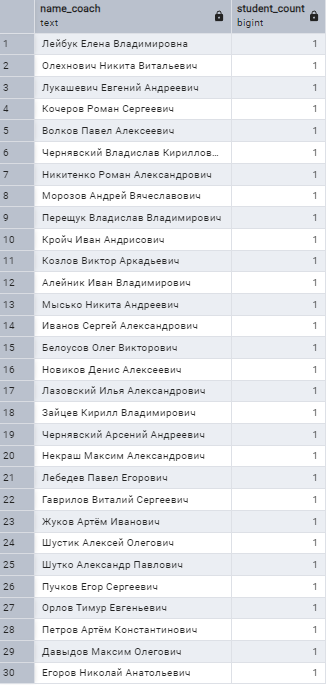


Рисунок 1.38 – Результат выполнения запроса

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы были изучены и применены на практике различные возможности языка SQL, включая использование подзапросов, агрегатных функций, группировку данных с оператором GROUP BY, а также операции над множествами (UNION, INTERSECT, EXCEPT). Эти инструменты позволили решить широкий спектр задач, связанных с выборкой, фильтрацией, обработкой и анализом данных из реляционных баз данных.

Подзапросами позволяют выполнять сложные вычисления и использовать промежуточные результаты для извлечения более точной информации. Агрегатные функции, такие как SUM, AVG, COUNT и другие, в сочетании с группировкой данных, продемонстрировали свою полезность при анализе больших массивов данных, предоставляя возможность получить ключевые метрики и характеристики. Также были изучены операции над множествами, которые являются эффективным инструментом для объединения, пересечения или вычитания данных из различных запросов.

Выполнение лабораторной работы способствовало не только закреплению теоретических знаний, но и развитию практических навыков работы с SQL. Полученные знания можно применять для решения реальных задач в области анализа данных, проектирования и оптимизации баз данных.