**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

A logo of a triangle

Description automatically generated

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Институт | компьютерных наук |
| Кафедра | автоматизированных систем управления |

ЛАБОРАТОРНАЯ РАБОТА №5

По дисциплине: "Базы данных"

На тему: "Проектирование запросов выборки данных"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ПИ-22-1 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Насонов Н.С. |
| Руководитель | |  |  |  |
| к.т.н., доцент  ученая степень, ученое звание | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Алексеев В.А. |

Липецк, 2024 г.

**Цель работы**

Изучить основы языка SQL, получить практические навыки разработки SELECT-запросов к базе данных с использованием конструкций селекции, сортировки, объединения, подзапросов, группировки данных и расчета статистических значений.

**Задание**

Разработать SELECT-запросы к БД, созданной в лабораторной работе №3, с использованием конструкций селекции, сортировки, объединения, подзапросов, группировки данных и расчета статистических значений. Проверить правильность работы запросов на тестовых данных.

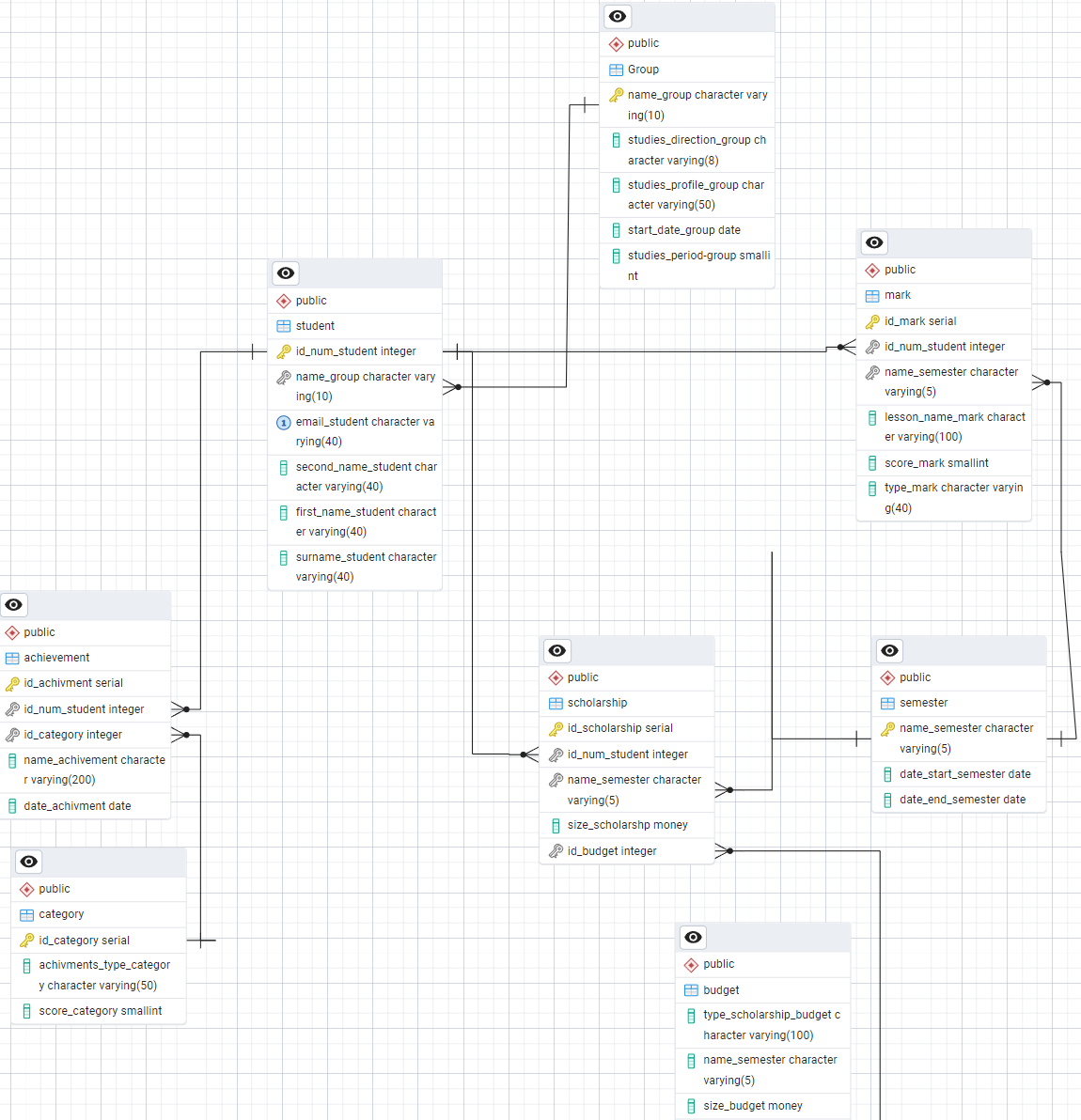


Рисунок 1 – ERD диаграмма

1. **Простой запрос выборки данных**

Получение групп, которые начали обучение в 2021-2022 году. Результат выполнения представлен на рисунке 2.

SELECT name\_group

FROM "group" g

WHERE extract(year from g.start\_date\_group) BETWEEN 2021 and 2022;

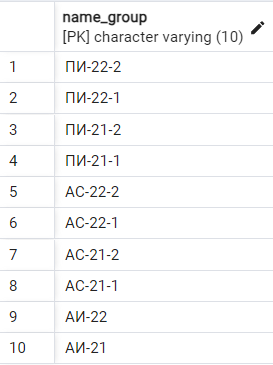


Рисунок 2 – Результат выполнения простого запроса

1. **Запросы выборки данных из нескольких таблиц**
   1. **Конструкция where**

Получение студентов за определённый семестр со стипендией в пределах 5000-6000. Результат выполнения представлен на рисунке 3.

SELECT s.first\_name\_student, s.second\_name\_student,

s.surname\_student, st.size\_scholarshp

FROM student s,scholarship st

WHERE st.id\_num\_student=s.id\_num\_student

and st.name\_semester='2023О'

and CAST(st.size\_scholarshp as numeric) BETWEEN 5000 and 6000;

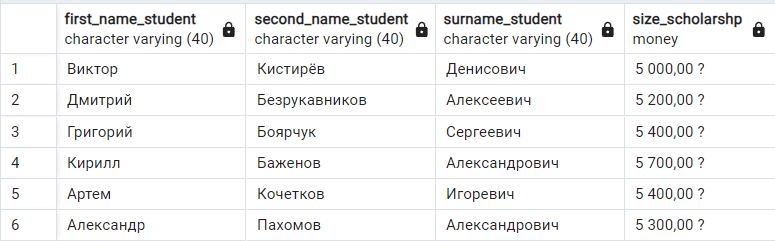


Рисунок 3 – Результат выполнения запроса по нескольким таблицам

* 1. **Внутреннее соединение**

Получение оценок студентов за определенный семестр. Результат выполнения представлен на рисунке 4.

SELECT s.first\_name\_student, s.second\_name\_student,

s.surname\_student,m.lesson\_name\_mark, m.score\_mark

FROM student s

INNER JOIN mark m ON s.id\_num\_student=m.id\_num\_student

WHERE m.name\_semester='2023В'

ORDER BY m.lesson\_name\_mark

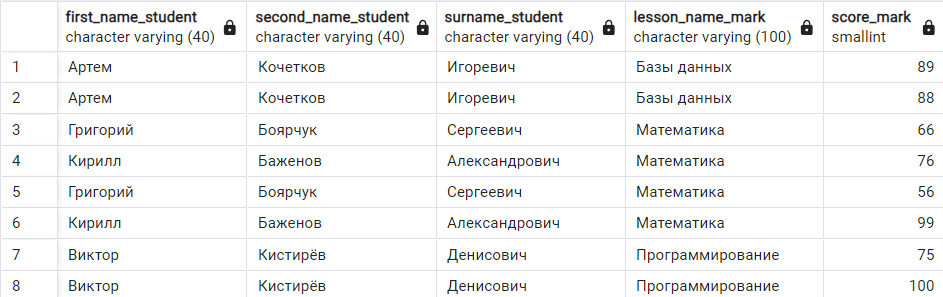


Рисунок 4 – Результат выполнения запроса с внутренним соединением

* 1. **Внешнее соединение**

Получение оценок студентов в том числе без оценок за какой-либо семестр. Результат выполнения представлен на рисунке 5.

SELECT s.first\_name\_student, s.second\_name\_student,

s.surname\_student,m.lesson\_name\_mark, m.score\_mark

FROM student s

LEFT JOIN mark m ON s.id\_num\_student=m.id\_num\_student

ORDER BY m.lesson\_name\_mark

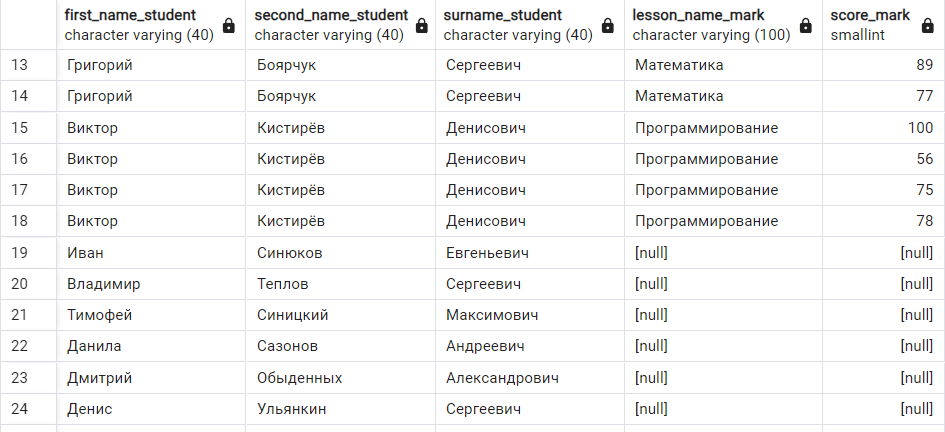


Рисунок 5 – Результат выполнения запроса с внешним соединением

1. **Запросы с подзапросом**
   1. **Скалярный подзапрос**

Получение среднего балла студента с id\_num\_student = 1220060404 и сравнение его с фиксированным значением. Результат выполнения представлен на рисунке 6.

SELECT

s.first\_name\_student,

s.second\_name\_student,

(SELECT AVG(m.score\_mark)

FROM mark m

WHERE m.id\_num\_student = 1220060404) AS avg\_mark

FROM student s

WHERE id\_num\_student = 1220060404;

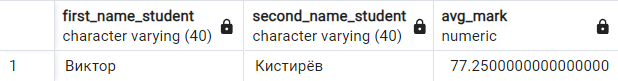


Рисунок 6 – Результат скалярного подзапроса

* 1. **Подзапрос по столбцу**

Выбрать студентов, у которых хотя бы одна оценка больше или равна 90. Результат выполнения представлен на рисунке 7.

SELECT \*

FROM student

WHERE id\_num\_student = ANY (

SELECT id\_num\_student

FROM mark

WHERE score\_mark >= 90

);

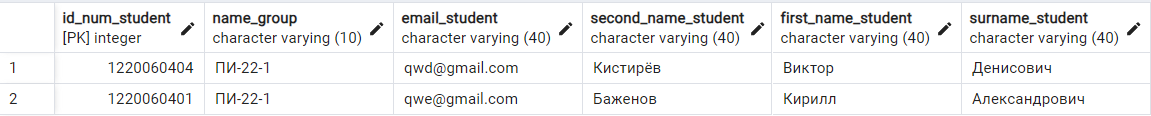


Рисунок 7 – Результат подзапроса по столбцу

* 1. **Табличный подзапрос**

Выбрать студентов, у которых есть хотя бы одна оценка выше 85. Результат выполнения представлен на рисунке 8.

SELECT \*

FROM student s

WHERE EXISTS (

SELECT 1

FROM mark m

WHERE m.id\_num\_student = s.id\_num\_student

AND m.score\_mark > 85

);

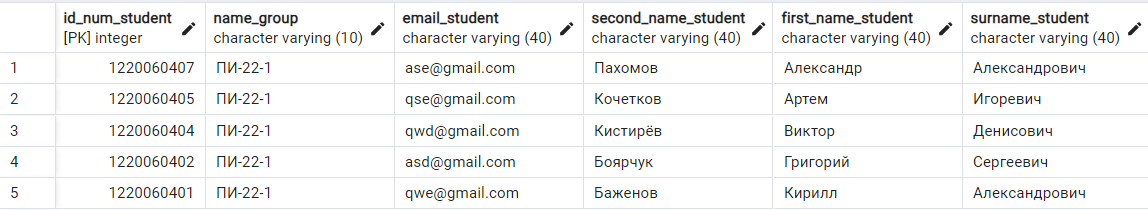


Рисунок 8 – Результат табличного подзапроса

1. **Запрос получения статических значений**
   1. **Расчёт итогов**

Количество студентов в каждой группе. Результат выполнения представлен на рисунке 9.

SELECT name\_group, COUNT(\*) AS students\_count

FROM student

GROUP BY name\_group;



Рисунок 9 – Результат выполнения запроса расчета итогов

* 1. **Расчет частных итогов с отбором по итоговым значениям**

Выбрать группы, в которых более 10 студентов. Результат выполнения представлен на рисунке 10.

SELECT name\_group, COUNT(\*) AS students\_count

FROM student

GROUP BY name\_group

HAVING COUNT(\*) > 10;

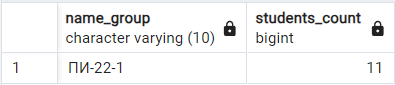


Рисунок 10 – Результат выполнения запроса расчета итогов

* 1. **Расчет частных итогов и итогов для каждой подкомбинации**

Общее количество студентов в каждой группе и общий итог. Результат выполнения представлен на рисунке 11.

SELECT name\_group, COUNT(\*) AS students\_count

FROM student

GROUP BY ROLLUP(name\_group);

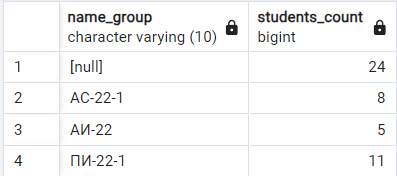


Рисунок 11 – Результат выполнения запроса расчета итогов

* 1. **Расчет частных итогов и итогов для каждой подкомбинации**

Подсчет суммы выданных стипендий для различных комбинаций бюджетов и семестров. Результат выполнения представлен на рисунке 12.

SELECT

name\_semester,

id\_budget,

SUM(size\_scholarshp) AS total\_scholarship

FROM scholarship

GROUP BY CUBE(name\_semester, id\_budget);

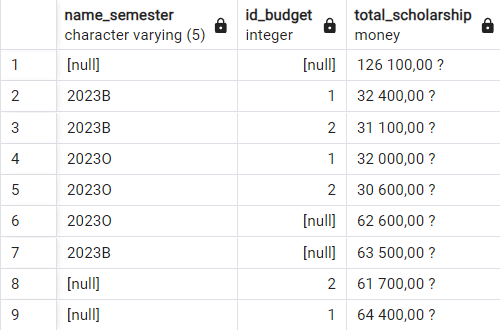


Рисунок 12 – Результат выполнения запроса расчета итогов

* 1. **Расчет итогов для каждого уникального значения каждого атрибута**

Подсчет достижений студентов отдельно по группам, категориям и годам достижения. Результат выполнения представлен на рисунке 13.

SELECT

name\_group,

id\_category,

date\_part('year', date\_achivment) AS achievement\_year,

COUNT(\*) AS achievement\_count

FROM achievement

JOIN student ON achievement.id\_num\_student = student.id\_num\_student

GROUP BY GROUPING SETS (

(name\_group),

(id\_category),

(date\_part('year', date\_achivment))

);

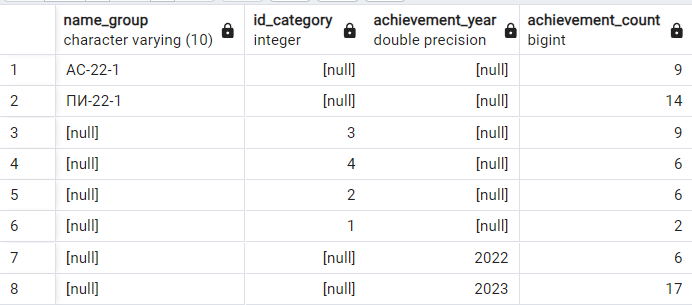


Рисунок 13 – Результат расчета итогов для каждого уникального значения

**Вывод**

В ходе выполнения лабораторной работы изучил работу с pgadmin. Создание и выполнение комплексных запросов SQL.