

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4
«Запросы на выборку и модификацию данных. Представления. Работа с
индексами»
по дисциплине «Проектирование и реализация баз данных»

Обучающийся Соболев Артём

Факультет прикладной информатики

Группа K3240

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург

2025

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

СХЕМА БАЗЫ ДАННЫХ

Схема логической модели базы данных:

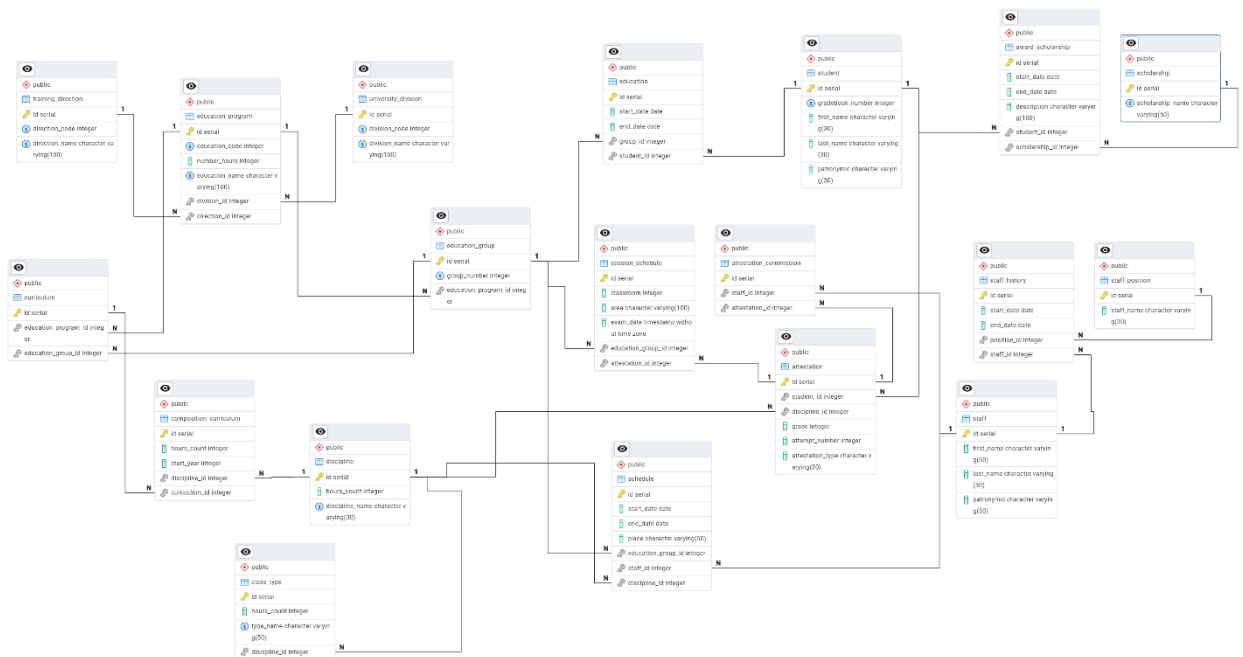


Рис. 1 – схема логической модели базы данных

ВЫПОЛНЕНИЕ

Запросы к базе данных

Задание: выполнить запросы согласно индивидуальному заданию, часть

2:

- Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей.
- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.
- Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине.
- Создать рейтинговый список групп по успеваемости по заданному направлению по результатам сдачи сессии, упорядочив его по убыванию.
- Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на

стипендии. Студент получает стипендию, если он сдал сессию без троек. Если студент не назначен на стипендию, указать 0, если назначен – 1.

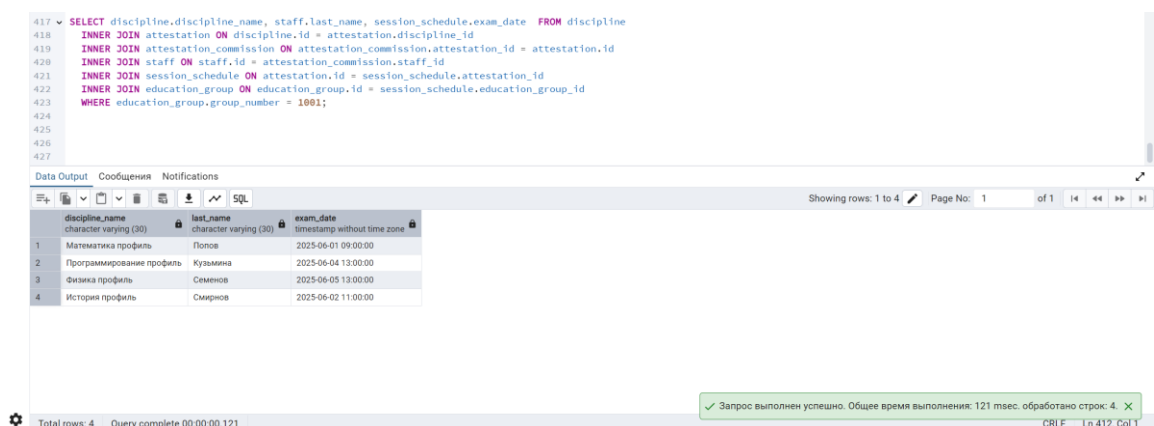
- Вывести список студентов, сдавших все положенные экзамены.
- Вывести список студентов, получивших максимальный средний балл в своей группе.

Выполнение

Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей.

Листинг 1 – SQL-запрос с получением списка дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей

```
SELECT      discipline.discipline_name,                staff.last_name,
session_schedule.exam_date FROM discipline
  INNER JOIN attestation ON discipline.id = attestation.discipline_id
  INNER JOIN attestation_commission ON attestation_commission.attestation_id =
attestation.id
  INNER JOIN staff ON staff.id = attestation_commission.staff_id
      INNER JOIN session_schedule ON attestation.id =
session_schedule.attestation_id
      INNER JOIN education_group ON education_group.id =
session_schedule.education_group_id
WHERE education_group.group_number = 1001;
```



The screenshot shows a database management tool interface. At the top, there is a text editor with the SQL query from the previous block. Below the editor, there is a table with the results of the query. The table has four columns: discipline_name, last_name, exam_date, and exam_date. The results show four rows of data for the group with group_number 1001.

discipline_name	last_name	exam_date	exam_date
Математика профиль	Попов	2025-06-01 09:00:00	
Программирование профиль	Кузмина	2025-06-04 13:00:00	
Физика профиль	Семенов	2025-06-05 13:00:00	
История профиль	Смирнов	2025-06-02 11:00:00	

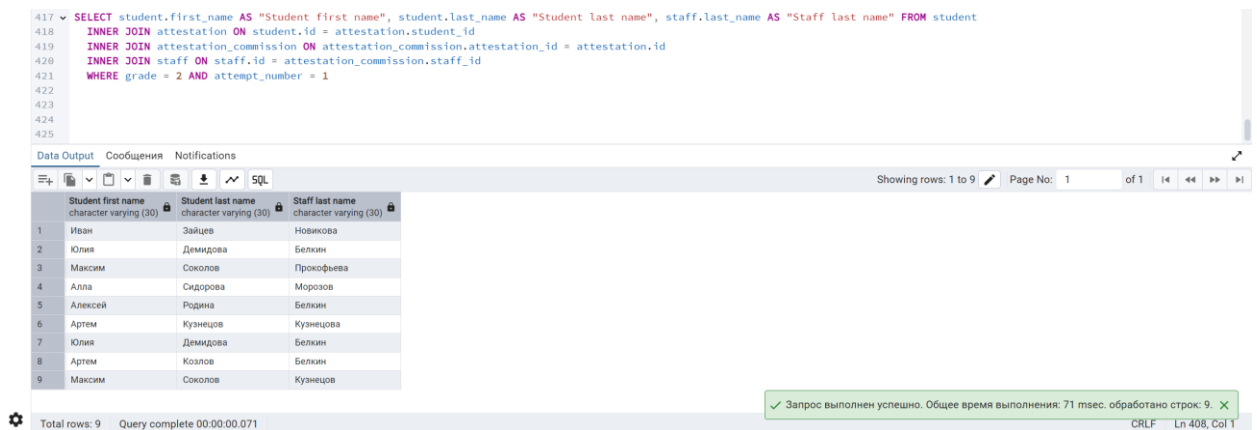
At the bottom of the screenshot, there is a status bar that says: "Запрос выполнен успешно. Общее время выполнения: 121 мсек. обработано строк: 4. X" (Query executed successfully. Total execution time: 121 msec. Rows processed: 4. X).

Рисунок 1 – результат выполнения SQL запроса

Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

Листинг 2 – SQL-запрос с получением списка студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен

```
SELECT student.first_name AS "Student first name", student.last_name AS "Student last name", staff.last_name AS "Staff last name" FROM student
  INNER JOIN attestation ON student.id = attestation.student_id
  INNER JOIN attestation_commission ON attestation_commission.attestation_id = attestation.id
  INNER JOIN staff ON staff.id = attestation_commission.staff_id
 WHERE grade = 2 AND attempt_number = 1
```



Student first name	Student last name	Staff last name
Иван	Зайцев	Новикова
Юлия	Демидова	Белкин
Максим	Соколов	Прокофьева
Алла	Сидорова	Морозов
Алексей	Родина	Белкин
Артем	Кузнецов	Кузнецова
Юлия	Демидова	Белкин
Артем	Козлов	Белкин
Максим	Соколов	Кузнецов

Рисунок 2 – результат выполнения SQL запроса

Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине.

Листинг 3 – SQL-запрос с получением фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине

```
SELECT DISTINCT(student.last_name) FROM discipline
  INNER JOIN attestation ON attestation.discipline_id = discipline.id
  INNER JOIN student ON attestation.student_id = student.id
 WHERE attestation.grade > (
   SELECT AVG(a2.grade)
   FROM attestation a2
   WHERE a2.discipline_id = attestation.discipline_id
 );
```

519 SELECT DISTINCT(student.last_name) FROM discipline
520 INNER JOIN attestation ON attestation.discipline_id = discipline.id
521 INNER JOIN student ON attestation.student_id = student.id
522 WHERE attestation.grade > (
523 SELECT AVG(a2.grade)
524 FROM attestation a2
525 WHERE a2.discipline_id = attestation.discipline_id
526);
527

Data OutputСообщенияNotifications

SQL

Showing rows: 1 to 51Page No: 1of 1

	last_name character varying (30)
1	Борисов
2	Борисова
3	Васильев
4	Гришин
5	Демидова
6	Дроздов
7	Дроздова
8	Жукова
9	Зайцев
10	Зайцева
11	Захарова
12	Козлов
13	Кузнецов

✓

Запрос выполнен успешно. Общее время выполнения: 113 мсек. обработано строк: 51.

✕

Total rows: 51Query complete 00:00:00.113CRLF Ln 515, Col 1

Рисунок 3 – результат выполнения SQL запроса

Создать рейтинговый список групп по успеваемости по заданному направлению по результатам сдачи сессии, упорядочив его по убыванию.

Листинг 4 – SQL-запрос с получением рейтингового списка групп по успеваемости по заданному направлению по результатам сдачи сессии, упорядочив его по убыванию

```

SELECT
    education_group.group_number AS "Группа",
    ROUND(AVG(attestation.grade), 2) AS "Средний балл"
FROM student
JOIN education ON education.student_id = student.id
JOIN education_group ON education.group_id = education_group.id
JOIN education_program ON education_group.education_program_id =
education_program.id
JOIN training_direction ON education_program.direction_id =
training_direction.id
JOIN attestation ON attestation.student_id = student.id
WHERE training_direction.direction_name = 'Гуманисты'
GROUP BY education_group.group_number
ORDER BY "Средний балл" DESC;

```

519 SELECT
520 education_group.group_number AS "Группа",
521 ROUND(AVG(attestation.grade), 2) AS "Средний балл"
522 FROM student
523 JOIN education ON education.student_id = student.id
524 JOIN education_group ON education_group.id = education.group_id
525 JOIN education_program ON education_group.education_program_id = education_program.id
526 JOIN training_direction ON education_program.direction_id = training_direction.id
527 JOIN attestation ON attestation.student_id = student.id
528 WHERE training_direction.direction_name = 'Гуманитаристы'
529 GROUP BY education_group.group_number
530 ORDER BY "Средний балл" DESC;
531

Data Output Сообщения Notifications

	Группа integer	Средний балл numeric
1	3002	4.50
2	3001	4.00
3	3003	4.00
4	3005	4.00
5	3004	3.50

Showing rows: 1 to 5 Page No: 1 of 1

Total rows: 5 Query complete 00:00:00.112 Запрос выполнен успешно. Общее время выполнения: 112 мсек. обработано строк: 5. CRLF Ln 513, Col 1

Рисунок 4 – результат выполнения SQL запроса

Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на стипендии. Студент получает стипендию, если он сдал сессию без троек. Если студент не назначен на стипендию, указать 0, если назначен – 1.

Листинг 5 – SQL-запрос с получением списка студентов, упорядоченных по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на стипендии

```
SELECT
    education_group.group_number AS "Группа",
    student.last_name AS "Фамилия",
    student.first_name AS "Имя",
    ROUND(AVG(attestation.grade), 2) AS "Средний балл",

    CASE
        WHEN MIN(attestation.grade) >= 4 AND EXISTS (
            SELECT 1 FROM award_scholarship
            WHERE award_scholarship.student_id = student.id
        ) THEN 1
        ELSE 0
    END AS "Стипендия"

FROM student
JOIN education ON education.student_id = student.id
JOIN education_group ON education_group.id = education.group_id
JOIN attestation ON attestation.student_id = student.id
```

```
GROUP BY education_group.group_number, student.last_name, student.first_name,
student.id
ORDER BY education_group.group_number, student.last_name, student.first_name;
```

```

1 SELECT
2   education_group.group_number AS "Группа",
3   student.last_name AS "Фамилия",
4   student.first_name AS "Имя",
5   ROUND(AVG(attestation.grade), 2) AS "Средний балл",
6
7   CASE
8     WHEN MIN(attestation.grade) >= 4 AND EXISTS (
9       SELECT 1 FROM award_scholarship
10      WHERE award_scholarship.student_id = student.id
11     ) THEN 1
12    ELSE 0
13   END AS "Стипендия"
14
15 FROM student
16 JOIN education ON education.student_id = student.id
17 JOIN education_group ON education_group.id = education.group_id
18 JOIN attestation ON attestation.student_id = student.id
19
20 GROUP BY education_group.group_number, student.last_name, student.first_name, student.id
21 ORDER BY education_group.group_number, student.last_name, student.first_name;
```

	Группа integer	Фамилия character varying (30)	Имя character varying (30)	Средний балл numeric	Стипендия integer
1	101	Демидова	Любовь	4.50	0
2	101	Дроздов	Артём	3.50	0
3	101	Кузнецова	Мария	4.50	0
4	101	Лапин	Роман	3.75	0
5	101	Лапин	Станислав	3.00	0
6	101	Родина	Алексей	3.67	0
7	101	Сафонова	Мария	4.00	1
8	101	Тихонова	Людмила	3.75	0
9	101	Фролова	Татьяна	3.80	0
10	102	Жукова	Алла	3.14	0
11	102	Жукова	Татьяна	3.75	0
12	102	Орлова	Наталья	3.29	0
13	102	Тихонов	Алексей	3.83	0
14	102	Тихонова	Татьяна	3.43	0
15	102	Федорова	Юлия	2.40	0

Total rows: 79 Query complete 00:00:00.099 Showing rows: 1 to 79 Page No: 1 of 1 CRLF Ln 6, Col 5

Запрос выполнен успешно. Общее время выполнения: 99 мсек. обработано строк: 79.

Рисунок 5 – результат выполнения SQL запроса

Вывести список студентов, сдавших все положенные экзамены.

Листинг 6 – SQL-запрос с получением списка студентов, сдавших все положенные экзамены

```

SELECT student.first_name, student.last_name FROM student
INNER JOIN (
  SELECT
    student.id, COUNT(DISTINCT discipline_id) AS passed_count FROM student
    INNER JOIN attestation ON student.id = attestation.student_id
    WHERE grade > 2
    GROUP BY student.id
) AS passed ON student.id = passed.id
INNER JOIN (
  SELECT
    student.id, COUNT(DISTINCT discipline_id) AS needed_count FROM student
    INNER JOIN education ON education.student_id = student.id
    INNER JOIN education_group ON education.group_id = education_group.id
    INNER JOIN curriculum ON curriculum.education_group_id =
    education_group.id
```



```

        INNER JOIN composition_curriculum ON
composition_curriculum.curriculum_id = curriculum.id
        GROUP BY student.id
    ) AS needed ON student.id = needed.id
WHERE passed.passed_count = needed.needed_count;

```

```

417 SELECT student.first_name, student.last_name FROM student
418     INNER JOIN (SELECT student.id AS student_id, COUNT(DISTINCT discipline_id) AS passed_count FROM student
419                 WHERE grade > 2
420                 GROUP BY student.id)
421     AS passed ON student.id = passed.student_id
422     INNER JOIN (SELECT student.id AS student_id, COUNT(DISTINCT discipline_id) AS needed_count FROM student
423                 WHERE grade < 2
424                 GROUP BY student.id)
425     AS needed ON student.id = needed.student_id
426     WHERE passed.passed_count = needed.needed_count;

```

	first_name character varying (30)	last_name character varying (30)
1	Иван	Иванов
2	Мария	Петрова
3	Алексей	Сидорова
4	Ольга	Кузнецова

Total rows: 4 Query complete 00:00:00.099 Запрос выполнен успешно. Общее время выполнения: 99 мсек. обработано строк: 4. CRLF Ln 414, Col 1

Рисунок 6 – результат выполнения SQL запроса

Вывести список студентов, получивших максимальный средний балл в своей группе.

Листинг 7 – SQL-запрос с получением списка студентов, получивших максимальный средний балл в своей группе

```

SELECT
    student_avg.first_name,
    student_avg.last_name,
    ROUND(student_avg.avg_grade, 2) AS avg_grade,
    student_avg.group_id
FROM (
    SELECT
        student.id,
        student.first_name,
        student.last_name,
        education.group_id,
        AVG(attestation.grade) AS avg_grade
    FROM student
    INNER JOIN attestation ON attestation.student_id = student.id
    INNER JOIN education ON education.student_id = student.id
    GROUP BY student.id, student.first_name, student.last_name,
education.group_id
) AS student_avg

```

```

INNER JOIN (
    SELECT
        group_id,
        MAX(avg_grade) AS max_avg
    FROM (
        SELECT
            student.id,
            education.group_id,
            AVG(attestation.grade) AS avg_grade
        FROM student
        INNER JOIN attestation ON attestation.student_id = student.id
        INNER JOIN education ON education.student_id = student.id
        GROUP BY student.id, education.group_id
    ) AS temp_avg
    GROUP BY group_id
) AS group_max_avg
ON student_avg.group_id = group_max_avg.group_id
AND student_avg.avg_grade = group_max_avg.max_avg;

```

91 SELECT
92 student_avg.first_name,
93 student_avg.last_name,
94 ROUND(student_avg.avg_grade, 2) AS avg_grade,
95 student_avg.group_id
96 FROM (
97 SELECT
98 student.id,
99 student.first_name,
100 student.last_name,
101 education.group_id,
102 AVG(attestation.grade) AS avg_grade
103 FROM student
104 INNER JOIN attestation ON attestation.student_id = student.id
105 INNER JOIN education ON education.student_id = student.id
106 GROUP BY student.id, student.first_name, student.last_name, education.group_id
107) AS student_avg
108 INNER JOIN (
109 SELECT
110 group_id,
111 MAX(avg_grade) AS max_avg

Data Output Сообщения Notifications

Showing rows: 1 to 22 Page No: 1 of 1

	first_name character varying (30)	last_name character varying (30)	avg_grade numeric	group_id integer
1	Алексей	Тихонов	3.83	2
2	Екатерина	Сафонова	4.25	4
3	Екатерина	Рябова	4.20	5
4	Дарья	Новикова	4.25	6
5	Сергей	Борисов	4.20	7
6	Екатерина	Новикова	4.40	9
7	Роман	Смирнов	4.20	10
8	Дмитрий	Лалин	4.25	3
9	Любовь	Демидова	4.50	1
10	Мария	Кузнецова	4.50	1
11	Людмила	Мельникова	4.75	8
12	Иван	Иванов	5.00	11
13	Мария	Петрова	5.00	11
14	Алексей	Сидоров	5.00	11
15	Ольга	Кузнецова	5.00	11
16	Павел	Никитин	5.00	17

Total rows: 22 Query complete 00:00:00.147

✓ Запрос выполнен успешно. Общее время выполнения: 147 msec. обработано строк: 22. ✕

CRLF Ln 93, Col 27

Рисунок 7 – результат выполнения SQL запроса

Представления

Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3:

- список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;
- данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

Выполнение

Список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

Листинг 8 – SQL-запрос с созданием представления списка студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен

```
CREATE VIEW students_with_two_grades AS
SELECT student.first_name AS student_first_name, student.last_name AS
student_last_name, staff.last_name AS staff_last_name FROM student
INNER JOIN attestation ON student.id = attestation.student_id
INNER JOIN attestation_commission ON attestation_commission.attestation_id =
attestation.id
INNER JOIN staff ON staff.id = attestation_commission.staff_id
WHERE grade = 2 AND attempt_number = 1;
```

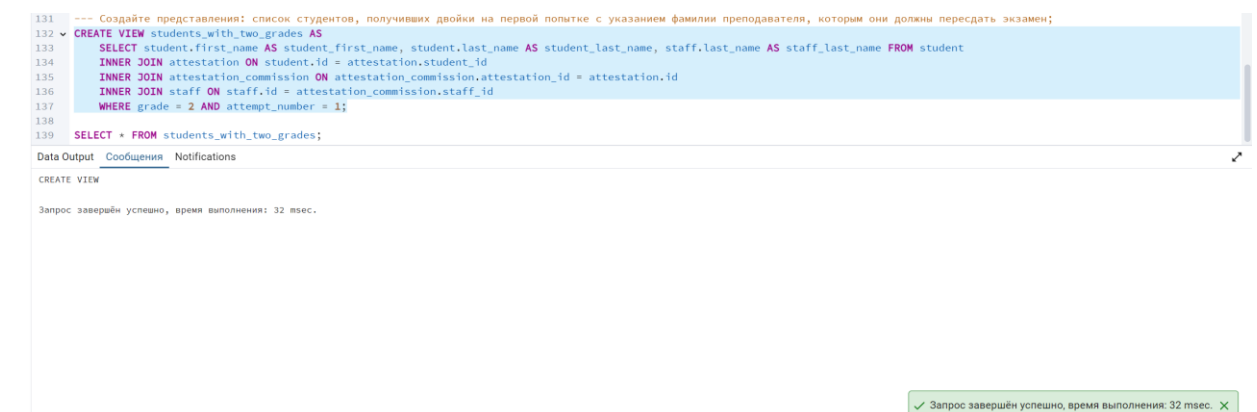


Рисунок 8 – результат выполнения SQL запроса

Листинг 9 – SQL-запрос с выводом представления

```
SELECT * FROM students_with_two_grades;
```

```

1311 --- Создайте представление: список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;
1312 CREATE VIEW students_with_two_grades AS
1313 SELECT student.first_name AS student_first_name, student.last_name AS student_last_name, staff.last_name AS staff_last_name FROM student
1314 INNER JOIN attestation ON student.id = attestation.student_id
1315 INNER JOIN attestation_commission ON attestation_commission.attestation_id = attestation.id
1316 INNER JOIN staff ON staff.id = attestation_commission.staff_id
1317 WHERE grade = 2 AND attempt_number = 1;
1318
1319 SELECT * FROM students_with_two_grades;

```

	student_first_name character varying (30)	student_last_name character varying (30)	staff_last_name character varying (30)
1	Мария	Сафонова	Морозов
2	Иван	Зайцев	Новикова
3	Юлия	Демидова	Белкин
4	Максим	Соколов	Прокофьева
5	Алла	Сидорова	Морозов
6	Алексей	Родина	Белкин
7	Артём	Кузнецов	Кузнецова
8	Юлия	Демидова	Белкин
9	Артём	Козлов	Белкин
10	Максим	Соколов	Кузнецов

Showing rows: 1 to 10 | Page No: 1 of 1

✓ Запрос выполнен успешно. Общее время выполнения: 34 мсек. обработано строк: 10. X

Total rows: 10 Query complete 00:00:00.034 CRLF Ln 139, Col 1

Рисунок 9 – результат выполнения SQL запроса

Данные о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

Листинг 10 – SQL-запрос с созданием представления данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки)

```

CREATE VIEW students_deduction AS
SELECT student.first_name AS student_first_name, student.last_name AS
student_last_name, staff.last_name AS staff_last_name FROM student
INNER JOIN attestation ON student.id = attestation.student_id
INNER JOIN attestation_commission ON attestation_commission.attestation_id =
attestation.id
INNER JOIN staff ON staff.id = attestation_commission.staff_id
WHERE grade = 2 AND attempt_number = 3;

```

```

132 --- Создайте представление: данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки)
133 CREATE VIEW students_deduction AS
134 SELECT student.first_name AS student_first_name, student.last_name AS student_last_name, staff.last_name AS staff_last_name FROM student
135 INNER JOIN attestation ON student.id = attestation.student_id
136 INNER JOIN attestation_commission ON attestation_commission.attestation_id = attestation.id
137 INNER JOIN staff ON staff.id = attestation_commission.staff_id
138 WHERE grade = 2 AND attempt_number = 3;
139
140 SELECT * FROM students_deduction;

```

CREATE VIEW

Запрос завершен успешно, время выполнения: 24 мсек.

✓ Запрос завершен успешно, время выполнения: 24 мсек. X

Total rows: Query complete 00:00:00.024 CRLF Ln 133, Col 1

Рисунок 10 – результат выполнения SQL запроса

Листинг 11 – SQL-запрос с выводом представления

```
SELECT * FROM students_deduction;
```

```
132 --- Создайте представление: данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки)
133 CREATE VIEW students_deduction AS
134 SELECT student.first_name AS student_first_name, student.last_name AS student_last_name, staff.last_name AS staff_last_name FROM student
135 INNER JOIN attestation ON student.id = attestation.student_id
136 INNER JOIN attestation_commission ON attestation_commission.attestation_id = attestation.id
137 INNER JOIN staff ON staff.id = attestation_commission.staff_id
138 WHERE grade = 2 AND attempt_number = 3;
139
140 SELECT * FROM students_deduction;
```

Data Output Сообщения Notifications

	student_first_name character varying (30)	student_last_name character varying (30)	staff_last_name character varying (30)
1	Артём	Коалов	Белкин

Showing rows: 1 to 1 Page No: 1 of 1 ⏪ ⏩ ⏴ ⏵

Total rows: 1 Query complete 00:00:00.080

✓ Запрос выполнен успешно. Общее время выполнения: 80 мсек. обработано строк: 1. ✕

CRLF Ln 140, Col 1

Рисунок 11 – результат выполнения SQL запроса

Запросы на модификацию данных

Задание: выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

INSEST запрос

Задание: назначить всем студентам, которые получили оценку «5» по всем положенным экзаменам, стипендию «ПГАС»

Листинг 12 – SQL-запрос с выводом назначенных стипендий «ПГАС»

```
SELECT first_name, last_name, gradebook_number, start_date, end_date
FROM award_scholarship
JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
JOIN student ON student.id = award_scholarship.student_id
WHERE scholarship_name = 'ПГАС';
```



Рисунок 12 – результат выполнения SQL запроса

Листинг 13 – SQL-запрос с вставкой данных

```
INSERT INTO award_scholarship (start_date, end_date, description, student_id,
scholarship_id)
SELECT
    '2023-01-01',
    '2023-06-01',
    'Стипендия ПГАС за все пятёрки',
    student.id,
    (SELECT scholarship.id FROM scholarship WHERE scholarship_name = 'ПГАС')
FROM student
WHERE student.id IN (
    SELECT passed_students.student_id
FROM (
```

```

        SELECT      student.id      AS      student_id,      COUNT(DISTINCT
attestation.discipline_id) AS passed_count
        FROM student
        JOIN attestation ON attestation.student_id = student.id
        WHERE attestation.grade = 5
        GROUP BY student.id
    ) AS passed_students
    JOIN (
        SELECT      student.id      AS      student_id,      COUNT(DISTINCT
composition_curriculum.discipline_id) AS needed_count
        FROM student
        JOIN education ON education.student_id = student.id
        JOIN education_group ON education.group_id = education_group.id
        JOIN curriculum ON curriculum.education_group_id = education_group.id
        JOIN composition_curriculum ON composition_curriculum.curriculum_id =
curriculum.id
        GROUP BY student.id
    ) AS      needed_students      ON      passed_students.student_id      =
needed_students.student_id
    WHERE passed_students.passed_count = needed_students.needed_count
)
AND NOT EXISTS (
    SELECT 1
    FROM award_scholarship
    JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
    WHERE award_scholarship.student_id = student.id
    AND scholarship.scholarship_name = 'ПГАС'
);

```

```

561 INSERT INTO award_scholarship (start_date, end_date, description, student_id, scholarship_id)
562 SELECT
563     '2023-01-01',|
564     '2023-06-01',
565     'Стипендия ПГАС за все пятёрки',
566     student_id,
567     (SELECT scholarship_id FROM scholarship WHERE scholarship_name = 'ПГАС')
568 FROM student
569 WHERE student_id IN (
570     SELECT passed_students.student_id
571     FROM (
572         SELECT student_id AS student_id, COUNT(DISTINCT attestation.discipline_id) AS passed_count
573         FROM student
574         JOIN attestation ON attestation.student_id = student.id
575         WHERE attestation.grade = 5
576         GROUP BY student_id
577     ) AS passed_students
578     JOIN (
579         SELECT student_id AS student_id, COUNT(DISTINCT composition_curriculum.discipline_id) AS needed_count
580         FROM student
581         JOIN education ON education.student_id = student_id
582         JOIN education_group ON education_group_id = education_group_id
583         JOIN curriculum ON curriculum.education_group_id = education_group_id
584         JOIN composition_curriculum ON composition_curriculum.curriculum_id = curriculum_id
585         GROUP BY student_id
586     ) AS needed_students ON passed_students.student_id = needed_students.student_id
587     WHERE passed_students.passed_count = needed_students.needed_count

```

Data Output Сообщения Notifications

INSERT в 24

Запрос завершён успешно, время выполнения: 30 msec.

Total rows: Query complete 00:00:00.030

✓ Запрос завершён успешно, время выполнения: 30 msec. X

CRLF Ln 563, Col 18

Рисунок 13 – результат выполнения SQL запроса

Листинг 14 – SQL-запрос с выводом назначенных стипендий «ПГАС»

```
SELECT first_name, last_name, gradebook_number, start_date, end_date
FROM award_scholarship
INNER JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
INNER JOIN student ON student.id = award_scholarship.student_id
WHERE scholarship_name = 'ПГАС';
```

119 SELECT first_name, last_name, gradebook_number, start_date, end_date
120 FROM award_scholarship
121 INNER JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
122 INNER JOIN student ON student.id = award_scholarship.student_id
123 WHERE scholarship_name = 'ПГАС';
124

	first_name character varying (30)	last_name character varying (30)	gradebook_number integer	start_date date	end_date date
1	Алла	Мельникова	991	2023-01-01	2023-06-01
2	Станислав	Лапин	1061	2023-01-01	2023-06-01
3	Александр	Сидоров	882	2023-01-01	2023-06-01
4	Вера	Тихонова	986	2023-01-01	2023-06-01
5	Дмитрий	Лапин	321	2023-01-01	2023-06-01
6	Григорий	Мельников	691	2023-01-01	2023-06-01
7	Дмитрий	Мельников	1051	2023-01-01	2023-06-01
8	Любовь	Демидова	468	2023-01-01	2023-06-01
9	Ольга	Кузнецова	1013	2023-01-01	2023-06-01
10	Любовь	Жукова	909	2023-01-01	2023-06-01
11	Роман	Васильев	608	2023-01-01	2023-06-01
12	Мария	Петрова	1011	2023-01-01	2023-06-01
13	Роман	Лапин	1063	2023-01-01	2023-06-01
14	Сергей	Сидоров	1059	2023-01-01	2023-06-01
15	Людмила	Новикова	703	2023-01-01	2023-06-01
16	Станислав	Лапин	124	2023-01-01	2023-06-01

Total rows: 24 Query complete 00:00:00.129

Запрос выполнен успешно. Общее время выполнения: 129 мс. обработано строк: 24.

Рисунок 14 – результат выполнения SQL запроса

UPDATE запрос

Задание: обновить у студента оценку по дисциплине на ППА

Листинг 15 – SQL-запрос с выводом оценки и номера попытки

```
SELECT attestation.grade, attestation.attempt_number FROM attestation
INNER JOIN student ON attestation.student_id = student.id
INNER JOIN discipline ON attestation.discipline_id = discipline.id
WHERE student.gradebook_number = 64 AND discipline.discipline_name =
'Экономика';
```

111 SELECT attestation.grade, attestation.attempt_number FROM attestation
112 INNER JOIN student ON attestation.student_id = student.id
113 INNER JOIN discipline ON attestation.discipline_id = discipline.id
114 WHERE student.gradebook_number = 64 AND discipline.discipline_name = 'Экономика';
115
116
117

	grade integer	attempt_number integer
1	2	1

Total rows: 1 Query complete 00:00:00.110

Запрос выполнен успешно. Общее время выполнения: 110 мс. обработано строк: 1.

Рисунок 15 – результат выполнения SQL запроса

Листинг 16 – SQL-запрос с вставкой данных

```
UPDATE attestation
SET
    grade = 3,
    attempt_number = attempt_number + 1
WHERE student_id = (
    SELECT id FROM student
    WHERE gradebook_number = 64
)
AND discipline_id = (
    SELECT id FROM discipline
    WHERE discipline_name = 'Экономика'
);
```

```
809 UPDATE attestation
810 SET
811     grade = 3,
812     attempt_number = attempt_number + 1
813 WHERE student_id = (
814     SELECT id FROM student
815     WHERE gradebook_number = 64
816 )
817 AND discipline_id = (
818     SELECT id FROM discipline
819     WHERE discipline_name = 'Экономика'
820 );
821
822
```

Data Output [Сообщения](#) [Notifications](#)

UPDATE 1

Запрос завершен успешно, время выполнения: 40 мсек.

Total rows: Query complete 00:00:00.040

✓ Запрос завершен успешно, время выполнения: 40 мсек. ✕

CRLF Ln 797, Col 1

Рисунок 16 – результат выполнения SQL запроса

Листинг 17 – SQL-запрос с выводом оценки и номера попытки

```
SELECT attestation.grade, attestation.attempt_number FROM attestation
INNER JOIN student ON attestation.student_id = student.id
INNER JOIN discipline ON attestation.discipline_id = discipline.id
WHERE student.gradebook_number = 64 AND discipline.discipline_name =
'Экономика';
```

```

99 SELECT attestation.grade, attestation.attempt_number FROM attestation
100 INNER JOIN student ON attestation.student_id = student.id
101 INNER JOIN discipline ON attestation.discipline_id = discipline.id
102 WHERE student.gradebook_number = 64 AND discipline.discipline_name = 'Экономика';
103

```

grade	attempt_number
3	2

Total rows: 1 Query complete 00:00:00.138

Запрос выполнен успешно. Общее время выполнения: 138 мсек. обработано строк: 1.

Рисунок 17 – результат выполнения SQL запроса

DELETE запрос

Задание: удалить у всех студентов стипендию «ПГАС», срок которой до сентября 2023 года или ранее

Листинг 18 – SQL-запрос с выводом студентов, у которых стипендия «ПГАС» закончилась в сентябре 2023 года

```

SELECT first_name, last_name, gradebook_number, end_date
FROM award_scholarship
INNER JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
INNER JOIN student ON student.id = award_scholarship.student_id
WHERE scholarship_name = 'ПГАС' AND award_scholarship.end_date < DATE '2023-09-01';

```

```

115 SELECT first_name, last_name, gradebook_number, end_date
116 FROM award_scholarship
117 INNER JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
118 INNER JOIN student ON student.id = award_scholarship.student_id
119 WHERE scholarship_name = 'ПГАС' AND award_scholarship.end_date < DATE '2023-09-01';
120

```

first_name	last_name	gradebook_number	end_date
Алла	Мельникова	991	2023-06-01
Станислав	Лапин	1061	2023-06-01
Александр	Сидоров	882	2023-06-01
Вера	Тихонова	386	2023-06-01
Дмитрий	Лапин	321	2023-06-01
Григорий	Мельников	691	2023-06-01
Дмитрий	Мельников	1051	2023-06-01
Любовь	Демидова	468	2023-06-01
Ольга	Кузнецова	1013	2023-06-01
Любовь	Жукова	909	2023-06-01
Роман	Васильев	608	2023-06-01
Мария	Петрова	1011	2023-06-01
Роман	Лапин	1063	2023-06-01
Сергей	Сидоров	1059	2023-06-01
Людмила	Новикова	703	2023-06-01
Станислав	Лапин	124	2023-06-01

Total rows: 24 Query complete 00:00:00.117

Запрос выполнен успешно. Общее время выполнения: 117 мсек. обработано строк: 24.

Рисунок 18 – результат выполнения SQL запроса

Листинг 19 – SQL-запрос с получением списка студентов, получивших максимальный средний балл в своей группе

```
DELETE FROM award_scholarship
WHERE scholarship_id = (
    SELECT id FROM scholarship WHERE scholarship_name = 'ПГАС'
)
AND end_date < DATE '2023-01-01';
```

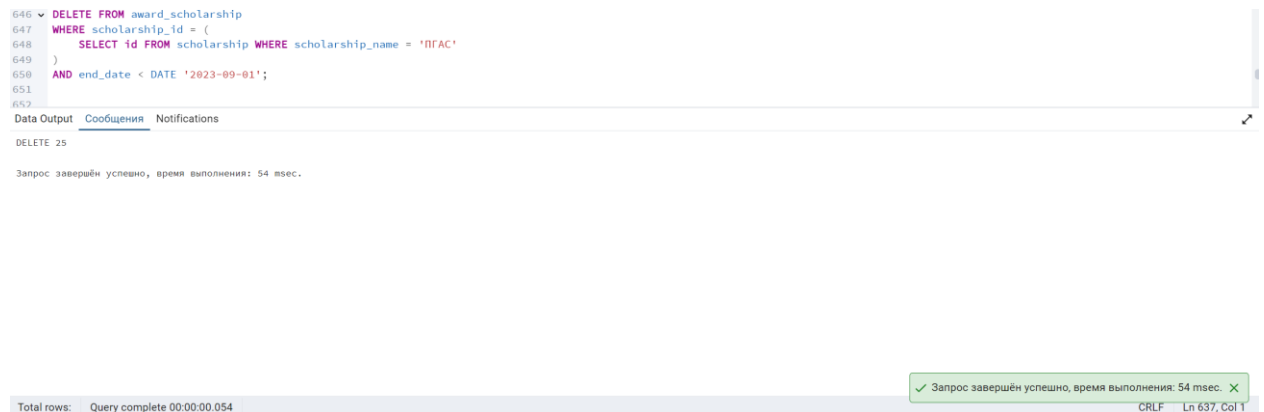


Рисунок 19 – результат выполнения SQL запроса

Листинг 20 – SQL-запрос с выводом студентов, у которых стипендия «ПГАС» закончилась в сентябре 2023 года

```
SELECT first_name, last_name, gradebook_number, end_date
FROM award_scholarship
INNER JOIN scholarship ON award_scholarship.scholarship_id = scholarship.id
INNER JOIN student ON student.id = award_scholarship.student_id
WHERE scholarship_name = 'ПГАС' AND award_scholarship.end_date < DATE '2023-09-01';
```

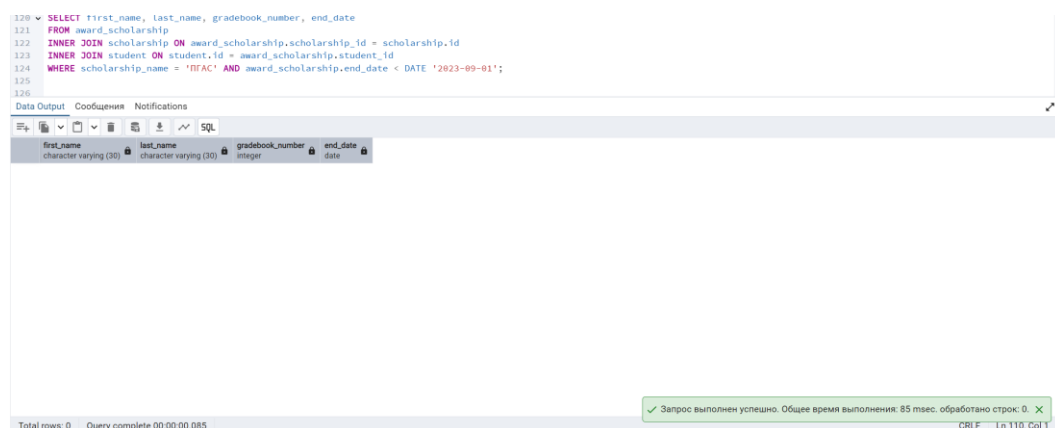


Рисунок 20 – результат выполнения SQL запроса

Создание индексов

Задание: выполнить запросы без индекса и создать планы запросов. Выполнить создание индексов. Выполнить запросы с индексами и создать планы запросов. Сравнить время выполнения запросов. Удалить индексы.

Анализ запроса для вывода информации о студентах из перечисленных имён или фамилий без индекса:

Листинг 21 – SQL запрос с анализом

```
EXPLAIN ANALYZE
SELECT *
FROM student
WHERE first_name IN ('Мария', 'Владислав', 'Иван', 'Егор', 'Дарья')
OR last_name IN ('Козлов', 'Фролов', 'Сафонова', 'Васильев');
```

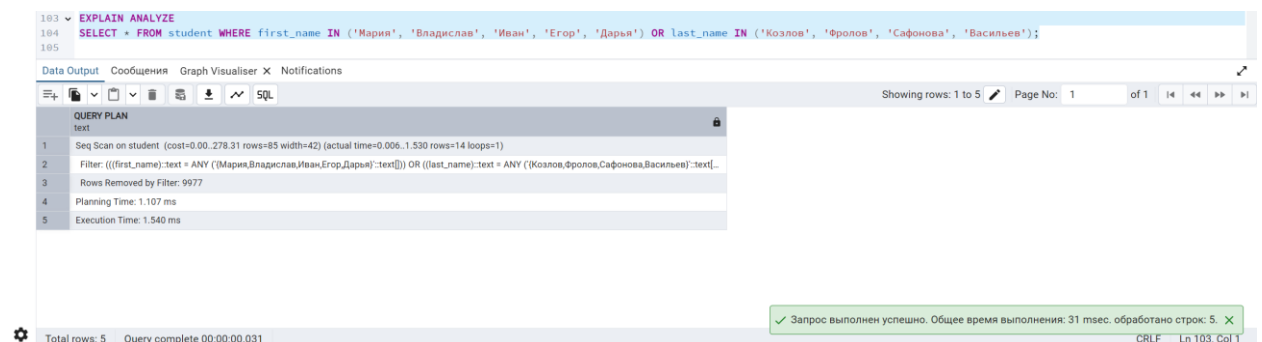


Рисунок 21 – результат выполнения SQL запроса

Время выполнения: *1.540 ms*

Листинг 22 – SQL-запрос с созданием индексов

```
CREATE INDEX student_first_name_idx ON student(first_name);
CREATE INDEX student_last_name_idx ON student(last_name);
```

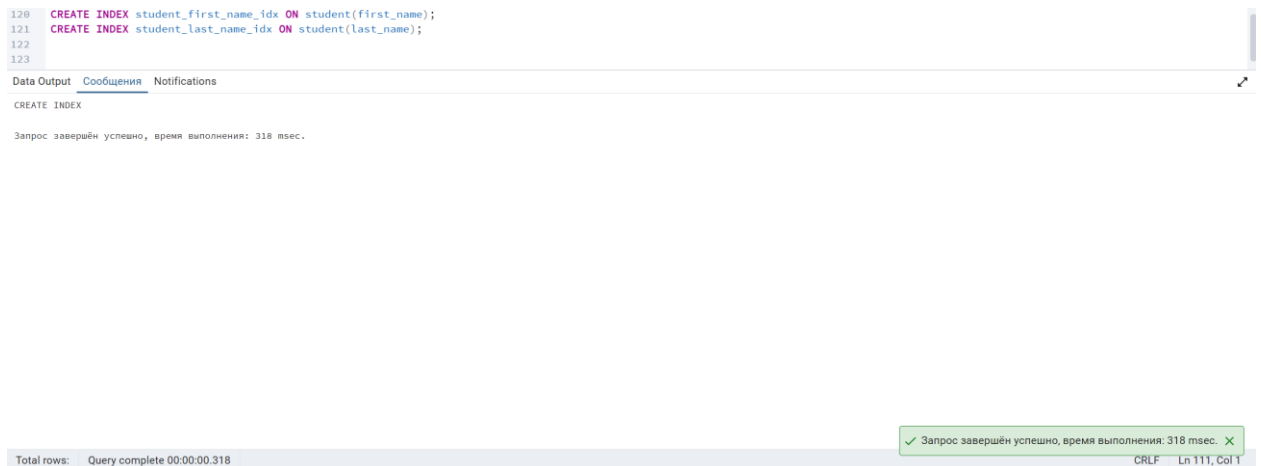


Рисунок 22 – результат выполнения SQL запроса

Анализ запроса для вывода информации о студентах из перечисленных имён или фамилий с индексом на поля *first_name* и *last_name* в таблице *student*:

Листинг 23 – SQL запрос с анализом

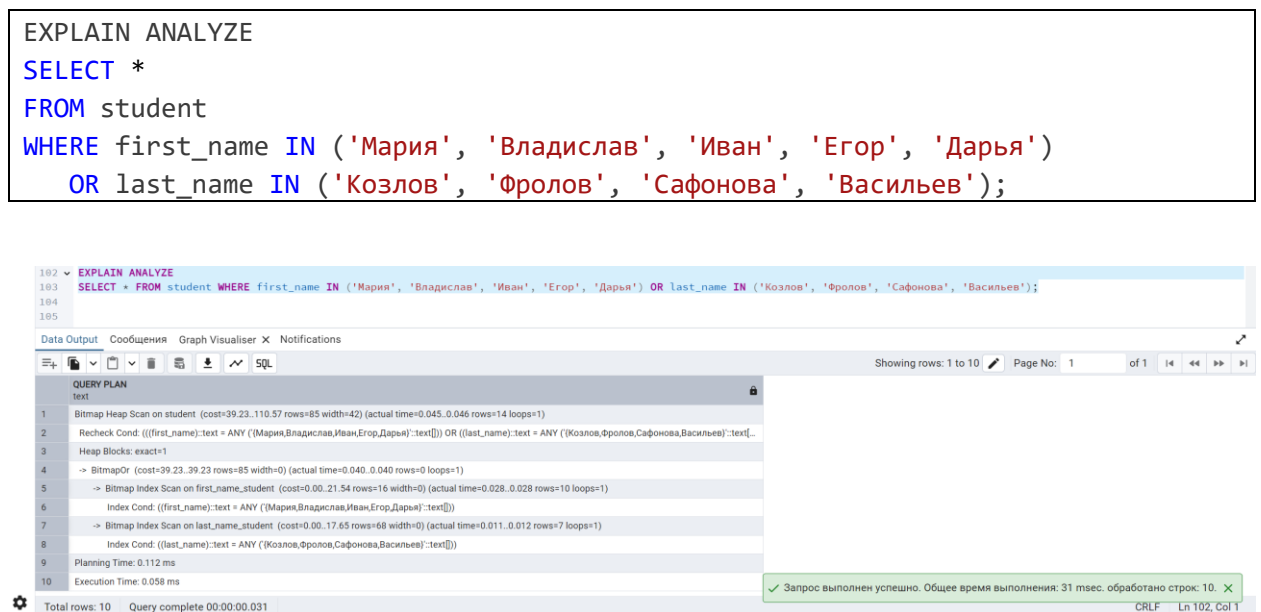
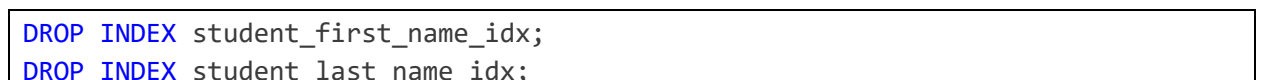


Рисунок 23 – результат выполнения SQL запроса

Время выполнения: *0.058 ms*. В 26 раз быстрее (см. рисунок 22)

Листинг 24 – SQL-запрос с удалением индексов



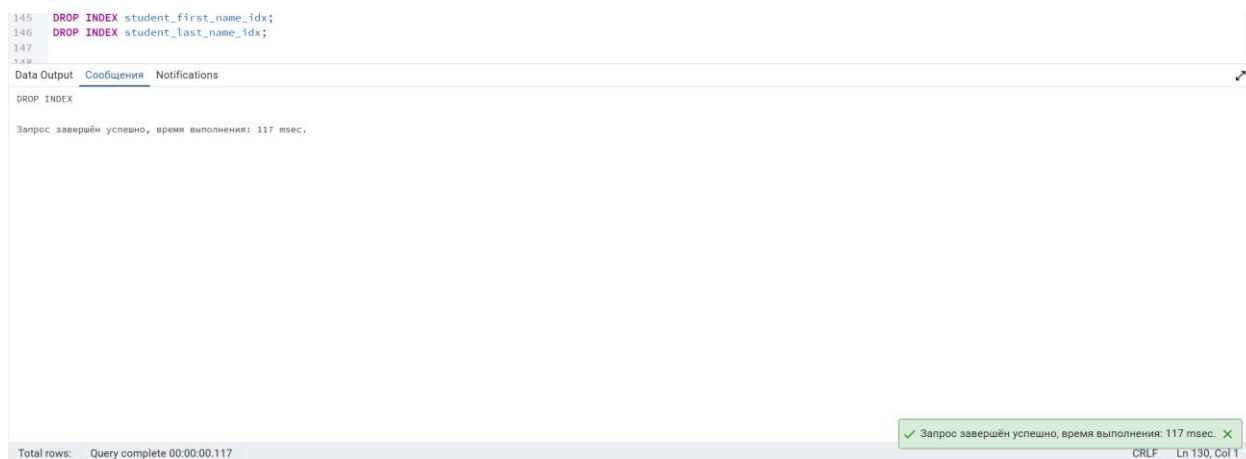


Рисунок 24 – результат выполнения SQL запроса

Вывод

При выполнении лабораторной работы были освоены практические навыки создания таблиц базы данных PostgreSQL, задание ограничений и связей между таблицами, заполнения их рабочими данными, резервного копирования и восстановления БД в pgAdmin.