

**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: Гайдук Алина Сергеевна
Факультет прикладной информатики
Группа K3241
Направление подготовки 09.03.03 Прикладная информатика
Образовательная программа Мобильные и сетевые технологии 2023
Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2025

Введение

Цель работы:

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Программное обеспечение:

СУБД PostgreSQL 1X, pgAdmin 4.

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

Модель БД представляет организацию сессии внутри университета, включает в себя информацию о преподавателях и их должностях, студентах, стипендии, дисциплинах и их типах, расписании, аттестационной комиссии и прочем. Подробнее ознакомиться со схемой базы данных можно на рисунке 1. Данная схема была создана в лабораторной работе №3. Для этого был использован генератор ERD-схемы в pgAdmin, настроено удобное отображение связей между таблицами.

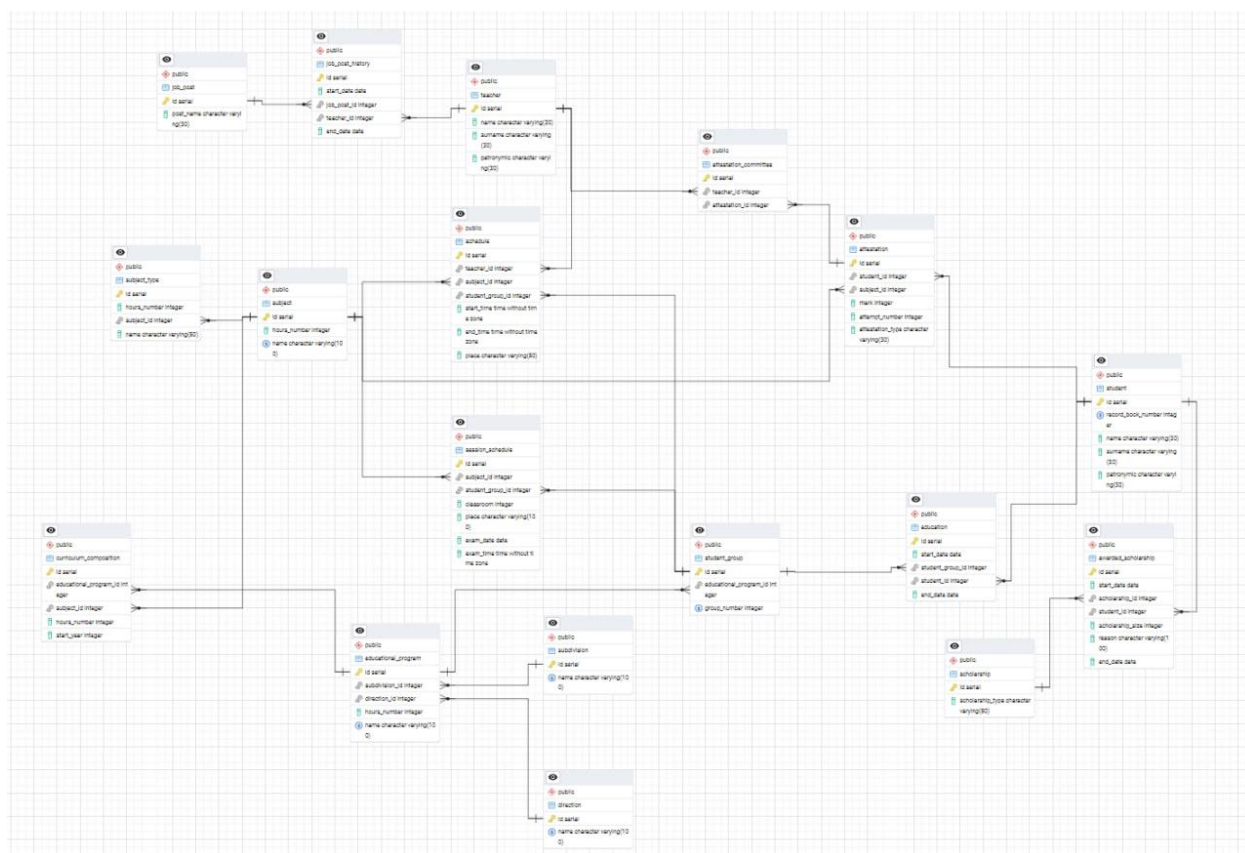


Рисунок 1 – Схема базы данных

Ход работы

1. Запросы к базе данных

Выполнить запросы согласно индивидуальному заданию, часть 2.

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

SQL-запрос, написанный для выполнения этого задания, представлен в листинге 1.

```
SELECT
subject.name,
teacher.surname,
session_schedule.exam_date
FROM subject
INNER JOIN attestation ON subject.id = attestation.subject_id
INNER JOIN attestation_committee ON attestation_committee.attestation_id =
attestation.id
INNER JOIN teacher ON teacher.id = attestation_committee.teacher_id
INNER JOIN session_schedule ON subject.id =
session_schedule.subject_id
INNER JOIN student_group ON student_group.id =
session_schedule.student_group_id
WHERE student_group.group_number = 103;
```

Листинг 1 – Запрос для создания списка дисциплин группы для сдачи

Результат выполнения данного SQL-запроса представлен на рисунке 2.



	name character varying (100)	surname character varying (30)	exam_date date
1	Линейная алгебра	Сидоров	2025-01-17

Рисунок 2 – Результат выполнения запроса

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

SQL-запрос, отвечающий заданному выше условию, представлен в листинге 2.

```

SELECT
student.name as student_name,
student.surname as student_surname,
student.patronymic as student_patronymic,
teacher.name as teacher_name,
teacher.surname as teacher_surname,
teacher.patronymic as teacher_patronymic
FROM attestation a
JOIN student ON student.id = a.student_id
JOIN schedule ON a.subject_id = schedule.subject_id
JOIN teacher ON schedule.teacher_id = teacher.id
WHERE a.mark = 2 AND a.attempt_number = 1

```

Листинг 2 – Запрос для создания списка студентов, получивших двойки на первой попытке

В результате выполнения запроса вывелась таблица, представленная на рисунке 3.

Data Output Сообщения Notifications						
Showing rows: 1 to 1 Page No: 1 of 1						
	student_name character varying (30)	student_surname character varying (30)	student_patronymic character varying (30)	teacher_name character varying (30)	teacher_surname character varying (30)	teacher_patronymic character varying (30)
1	Николай	Попов	Алексеевич	Мария	Дроздова	Павловна

Рисунок 3 – Результат выполнения запроса

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

Запрос, созданный для решения данного задания, представлен в листинге 3. Он включает в себя подзапрос с вычислением среднего балла по дисциплине.

```

SELECT
student.surname as student_surname,
student.name as student_name,
subject.name as subject_name,
a.mark as student_mark
FROM attestation a
JOIN student ON a.student_id = student.id
JOIN subject ON a.subject_id = subject.id
WHERE a.mark > (
SELECT AVG(a2.mark)
FROM attestation a2
WHERE a2.subject_id = a.subject_id
)

```

Листинг 3 – Запрос для нахождения студентов, имеющих балл выше среднего

В качестве результата я получила таблицу, которая представлена на рисунке 4.

Data Output Сообщения Notifications				
Showing rows: 1				
	student_surname character varying (30)	student_name character varying (30)	subject_name character varying (100)	student_mark integer
1	Сидоров	Алексей	Программирование	5
2	Смирнов	Дмитрий	Программирование	5
3	Сергеев	Павел	Программирование	5

Рисунок 4 – Результат выполнения запроса

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

Для создания рейтинга групп по успеваемости был написан SQL-запрос, представленный в листинге 4.

```
SELECT
    student_group.group_number as group_number,
    ROUND(AVG(a.mark), 2) as average_mark
FROM student_group
JOIN education ON education.student_group_id = student_group.id
JOIN attestation a ON a.student_id = education.student_id
JOIN educational_program ep ON student_group.educational_program_id = ep.id
JOIN direction d ON ep.direction_id = d.id
WHERE d.id = 1
GROUP BY student_group.id, student_group.group_number
ORDER BY average_mark DESC;
```

Листинг 4 – Запрос для создания рейтинга групп по успеваемости

В результате я получила таблицу (см. рис. 5).

Data Output Сообщения Notifications		
Showing rows: 1		
	group_number integer	average_mark numeric
1	101	5.00
2	153	4.25
3	152	3.80
4	151	3.40

Рисунок 5 – Результат выполнения 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.

Для выполнения данного задания был написан SQL-запрос, он представлен в листинге 5.

```
SELECT
  student_group.group_number as student_group,
  student.surname as student_surname,
  student.name as student_name,
  ROUND(AVG(a.mark), 2) as average_mark,
  CASE
    WHEN MIN(a.mark) > 3 THEN
      CASE
        WHEN awarded_scholarship.id IS NOT NULL THEN 1
        ELSE 0
      END
    ELSE 0
  END AS has_scholarship
FROM student
JOIN education ON education.student_id = student.id
JOIN student_group ON student_group.id = education.student_group_id
JOIN attestation a ON a.student_id = student.id
LEFT JOIN awarded_scholarship ON awarded_scholarship.student_id = student.id
GROUP BY student.id, student.surname, student.name,
  student_group.group_number, awarded_scholarship.id
ORDER BY student_group.group_number, student.surname, student.name;
```

Листинг 5 – Запрос для создания списка студентов, среднего балла и стипендии

Таблица, полученная в результате запроса выше, представлена на рисунке 6.

Data Output Сообщения Notifications						
SQL						
	student_group integer	student_surname character varying (30)	student_name character varying (30)	average_mark numeric	has_scholarship integer	
1	101	Сидоров	Алексей	5.00	1	
2	102	Кузнецова	Мария	4.00	1	
3	103	Новиков	Олег	3.00	0	
4	104	Андреев	Андрей	5.00	1	
5	105	Смирнов	Дмитрий	4.50	1	
6	106	Кузнецов	Сергей	3.00	0	
7	107	Попов	Николай	3.00	0	
8	108	Лебедев	Владимир	4.50	1	
9	109	Новиков	Михаил	4.00	1	
10	110	Морозов	Евгений	3.00	0	
11	111	Соболев	Александр	2.00	0	
12	112	Павлов	Максим	5.00	1	
13	113	Козлов	Артем	4.00	1	
14	114	Васильев	Роман	3.00	0	
15	115	Зайцев	Илья	4.00	0	
16	151	Горшков	Константин	4.00	0	
17	151	Романов	Виктор	2.50	0	
18	152	Васильевский	Денис	3.50	0	
19	152	Кузьмин	Кирилл	4.00	0	
20	153	Сергеев	Павел	4.25	0	

Рисунок 6 – Результат выполнения запроса

- Вывести список студентов, сдавших все положенные экзамены.

Для создания такого списка я написала SQL-запрос к базе данных, он представлен в листинге 6.

```
SELECT
    student.surname as student_surname,
    student.name as student_name,
    student_group.group_number
FROM student JOIN education ON education.student_id = student.id
JOIN student_group ON student_group.id = education.student_group_id
WHERE NOT EXISTS (
    SELECT 1
    FROM curriculum_composition
    LEFT JOIN attestation
        ON attestation.subject_id = curriculum_composition.subject_id
        AND attestation.student_id = student.id
    WHERE curriculum_composition.educational_program_id =
        student_group.educational_program_id
        AND (attestation.mark IS NULL OR attestation.mark <= 2)
)
ORDER BY student_group.group_number, student.surname;
```

Листинг 6 – Запрос для создания списка студентов, сдавших экзамены

Полученная таблица представлена на рисунке 7.

Data Output Сообщения Notifications			
	student_surname character varying (30)	student_name character varying (30)	group_number integer
1	Соболев	Александр	111
2	Павлов	Максим	112
3	Козлов	Артем	113
4	Васильев	Роман	114
5	Зайцев	Илья	115

Рисунок 7 – Результат выполнения запроса

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

Для вывода отличившихся студентов был написан SQL-запрос, представленный в листинге 7.

```
SELECT
    student.surname AS student_surname,
    student.name AS student_name,
    student_group.group_number,
    ROUND(AVG(attestation.mark), 2) AS average_mark
FROM student
    JOIN education
        ON education.student_id = student.id
    JOIN student_group
        ON student_group.id = education.student_group_id
    JOIN attestation
        ON attestation.student_id = student.id
GROUP BY student.id, student.surname, student.name,
    student_group.group_number
HAVING ROUND(AVG(attestation.mark), 2) = (
    SELECT ROUND(MAX(avg_mark), 2)
    FROM (
        SELECT
            AVG(attestation.mark) AS avg_mark
        FROM student
            JOIN education
                ON education.student_id = student.id
            JOIN student_group
                ON student_group.id = education.student_group_id
            JOIN attestation
                ON attestation.student_id = student.id
        WHERE student_group.group_number = student_group.group_number
```

```

GROUP BY student.id
) AS group_avg
)
ORDER BY student_group.group_number, student.surname;

```

Листинг 7 – Запрос для создания списка отличников

Была получена таблица, представленная на рисунке 8.

Data Output Сообщения Notifications				
Showing				
	student_surname character varying (30)	student_name character varying (30)	group_number integer	average_mark numeric
1	Сидоров	Алексей	101	5.00
2	Андреев	Андрей	104	5.00
3	Павлов	Максим	112	5.00

Рисунок 8 – Результат выполнения запроса

2. Представления

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

- Создать список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;

Я создала представление, соответствующее условиям данного задания (см. листинг 8).

```

CREATE OR REPLACE VIEW bad_students AS
SELECT
    student.surname AS student_surname,
    student.name AS student_name,
    subject.name AS subject_name,
    teacher.surname AS teacher_surname
FROM attestation
JOIN student ON student.id = attestation.student_id
JOIN subject ON subject.id = attestation.subject_id
JOIN attestation_committee ON attestation_committee.attestation_id =
attestation.id
JOIN teacher ON teacher.id = attestation_committee.teacher_id
WHERE attestation.mark = 2 AND attestation.attempt_number = 1
ORDER BY student.surname, subject.name;

```

Листинг 8 – Создание представления списка студентов-двоечников

Результат выполнения данного запроса представлен на рисунке 9.

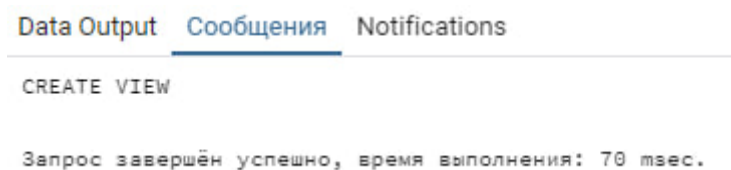


Рисунок 9 – Результат выполнения запроса

Для того, чтобы просмотреть содержимое представления, я написала SQL-запрос, представленный в листинге 9.

```
SELECT * from bad_students
```

Листинг 9 – Запрос для просмотра содержимого представления

В результате я получила таблицу, представленную на рисунке 10.

	student_surname character varying (30)	student_name character varying (30)	subject_name character varying (100)	teacher_surname character varying (30)
1	Васильевский	Денис	Структуры данных	Попов
2	Васильевский	Денис	Структуры данных	Смирнова
3	Васильевский	Денис	Структуры данных	Дроздова
4	Козлов	Артем	Линейная алгебра	Смирнова
5	Козлов	Артем	Линейная алгебра	Попов
6	Козлов	Артем	Линейная алгебра	Соболев
7	Логина	Алёна	Численные методы	Старкова
8	Морозова	Виктория	Теория вероятностей	Попов
9	Морозова	Виктория	Теория вероятностей	Иванов
10	Попов	Николай	Структуры данных	Смирнова
11	Попов	Николай	Структуры данных	Дроздова
12	Попов	Николай	Структуры данных	Попов

Рисунок 10 – Результат выполнения запроса

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

В листинге 10 представлен запрос для создания представления по условиям данного задания.

```

CREATE VIEW bad_students2 AS
SELECT
    student.surname AS student_surname,
    student.name AS student_name,
    subject.name AS subject_name,
    attestation.mark,
    attestation.attempt_number,
    teacher.surname AS teacher_surname
FROM attestation
JOIN student ON student.id = attestation.student_id
JOIN subject ON subject.id = attestation.subject_id
JOIN attestation_committee ON attestation_committee.attestation_id =
attestation.id
JOIN teacher ON teacher.id = attestation_committee.teacher_id
WHERE attestation.mark = 2 AND attestation.attempt_number >= 3
ORDER BY student.surname, subject.name;

```

Листинг 10 – Запрос для создания представления списка студентов-двоечников (после 3-й попытки)

Результат выполнения данного запроса представлен на рисунке 11.

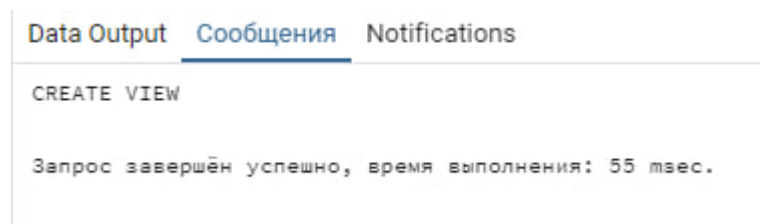


Рисунок 11 – Результат выполнения запроса

Для того, чтобы просмотреть содержимое представления, я написала SQL-запрос, представленный в листинге 11.

```

SELECT * from bad_students2

```

Листинг 9 – Запрос для просмотра содержимого представления

В результате я получила таблицу, представленную на рисунке 12.

	student_surname character varying (30)	student_name character varying (30)	subject_name character varying (100)	mark integer	attempt_number integer	teacher_surname character varying (30)
1	Романов	Виктор	Компьютерные сети	2	3	Смирнова
2	Соболев	Александр	Компьютерные сети	2	3	Попов

Рисунок 12 – Результат выполнения запроса

3. Запросы на модификацию данных

Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, SQL-команду, скриншот до и после выполнения запроса с результатом (для сравнения). Учитывается сложность запроса;

3.1 INSERT

- Создать расписание пересдачи для студентов, получивших двойку.

Для начала необходимо посмотреть, имеются ли уже какие-то записи в расписании аттестации для студентов с двойками на первой попытке. Для этого я написала запрос, представленный в листинге 12.

```
SELECT
    subject.name AS subject_name,
    student_group.group_number AS group_number,
    session_schedule.classroom,
    session_schedule.place,
    session_schedule.exam_date,
    session_schedule.exam_time
FROM session_schedule
JOIN subject
    ON subject.id = session_schedule.subject_id
JOIN student_group
    ON student_group.id = session_schedule.student_group_id
WHERE (session_schedule.subject_id, session_schedule.student_group_id) IN (
    SELECT
        attestation.subject_id,
        education.student_group_id
    FROM attestation
    JOIN education
        ON education.student_id = attestation.student_id
    WHERE attestation.mark = 2
)
ORDER BY session_schedule.exam_date, subject.name;
```

Листинг 12 – Запрос для просмотра расписания аттестаций двоечников

В результате выполнения запроса вывелась непустая таблица, представленная на рисунке 13. Наличие данных в ней говорит о том, что ранее я вручную добавила пересдачи для двоечников.

Data Output Сообщения Notifications						
Showing rows: 1 to 15 Page No: 1						
	subject_name character varying (100)	group_number integer	classroom integer	place character varying (100)	exam_date date	exam_time time without time zone
1	Математический анализ	102	202	Корпус В	2025-01-16	11:00:00
2	Линейная алгебра	103	103	Корпус А	2025-01-17	12:00:00
3	Дискретная математика	104	204	Корпус В	2025-01-18	13:00:00
4	Теория графов	114	214	Корпус В	2025-01-28	11:00:00

Рисунок 13 – Результат выполнения запроса

Теперь нужно сделать единое расписание для всех студентов, сдавших экзамен на «2». Для этого я прописала запрос, показанный в листинге 13.

```

INSERT INTO session_schedule (subject_id, student_group_id, classroom,
place, exam_date, exam_time)
SELECT
    attestation.subject_id,
    education.student_group_id,
    CASE
        WHEN attestation.subject_id % 2 = 0 THEN 301
        ELSE 302
    END AS classroom,
    'Корпус В' AS place,
    CURRENT_DATE + INTERVAL '14 days' AS exam_date,
    '10:00' AS exam_time
FROM attestation
JOIN student ON student.id = attestation.student_id
JOIN education ON education.student_id = student.id
JOIN subject ON subject.id = attestation.subject_id
WHERE attestation.mark = 2;

```

Листинг 13 – Запрос для внесения информации о расписании пересдач для студентов, сдавших экзамен на «2»

Снова прописав запрос, представленный в листинге 12, получаем таблицу, показанную на рисунке 14. В исходную таблицу добавились записи единого расписания пересдач, аудитория высчитывается с идентификатора аттестации.

Data Output Сообщения Notifications							
Showing rows: 1 to 15 Page No: 1							
	subject_name character varying (100)	group_number integer	classroom integer	place character varying (100)	exam_date date	exam_time time without time zone	
1	Математический анализ	102	202	Корпус В	2025-01-16	11:00:00	
2	Линейная алгебра	103	103	Корпус А	2025-01-17	12:00:00	
3	Дискретная математика	104	204	Корпус В	2025-01-18	13:00:00	
4	Теория графов	114	214	Корпус В	2025-01-28	11:00:00	
5	Дискретная математика	104	301	Корпус В	2025-05-11	10:00:00	
6	Компьютерные сети	151	301	Корпус В	2025-05-11	10:00:00	
7	Компьютерные сети	111	301	Корпус В	2025-05-11	10:00:00	
8	Линейная алгебра	113	302	Корпус В	2025-05-11	10:00:00	
9	Линейная алгебра	103	302	Корпус В	2025-05-11	10:00:00	
10	Математический анализ	102	301	Корпус В	2025-05-11	10:00:00	
11	Структуры данных	107	301	Корпус В	2025-05-11	10:00:00	
12	Структуры данных	152	301	Корпус В	2025-05-11	10:00:00	
13	Теория вероятностей	110	302	Корпус В	2025-05-11	10:00:00	
14	Теория графов	114	301	Корпус В	2025-05-11	10:00:00	
15	Численные методы	112	302	Корпус В	2025-05-11	10:00:00	

Рисунок 14 – Результат выполнения запроса

3.2 UPDATE

- Повысить на 2000 рублей стипендию студентам, чей средний балл по всем предметам выше 4.5.

Для начала необходимо просмотреть стипендии и средний балл студентов, для этого я написала запрос, представленный в листинге 14.

```
SELECT
    student.surname AS student_surname,
    student.name AS student_name,
    awarded_scholarship.scholarship_size,
    ROUND(AVG(attestation.mark), 2) AS average_mark
FROM awarded_scholarship
JOIN student ON student.id = awarded_scholarship.student_id
JOIN attestation ON attestation.student_id = student.id
GROUP BY student.id, student.surname, student.name,
awarded_scholarship.scholarship_size
ORDER BY average_mark DESC;
```

Листинг 14 – Просмотр студентов, их среднего балла и стипендии

В результате выполнения запроса получаю таблицу, представленную на рисунке 15. Стоит отметить, что студенты со средним баллом меньше 4, не получают никакую стипендию.

	student_surname character varying (30)	student_name character varying (30)	scholarship_size integer	average_mark numeric
1	Павлов	Максим	4600	5.00
2	Сидоров	Алексей	5000	5.00
3	Андреев	Андрей	3500	5.00
4	Лебедев	Владимир	2500	4.50
5	Смирнов	Дмитрий	4000	4.50
6	Сергеев	Павел	4100	4.25
7	Зайцев	Илья	4200	4.00
8	Новиков	Михаил	2000	4.00
9	Кузьмин	Кирилл	3900	4.00
10	Горшков	Константин	3700	4.00
11	Кузнецова	Мария	3000	4.00
12	Васильевский	Денис	0	3.50
13	Попов	Николай	0	3.00
14	Кузнецов	Сергей	0	3.00
15	Романов	Виктор	0	3.00
16	Морозов	Евгений	0	3.00
17	Козлов	Артем	0	3.00
18	Новиков	Олег	0	3.00
19	Васильев	Роман	0	2.50

Рисунок 15 – Результат выполнения запроса

Далее прописываю UPDATE-запрос, который обновляет стипендию студентам с высоким средним баллом. Запрос представлен в листинге 15.

```
UPDATE awarded_scholarship
SET scholarship_size = scholarship_size + 2000
WHERE student_id IN (
    SELECT student_id
    FROM attestation
    GROUP BY student_id
    HAVING AVG(mark) > 4.5
);
```

Листинг 15 – Запрос для обновления стипендии

Далее снова прописываем запрос, указанный в листинге 14, чтобы посмотреть новые стипендии студентов. Результат выполнения запроса представлен на рисунке 16.

	student_surname character varying (30)	student_name character varying (30)	scholarship_size integer	average_mark numeric
1	Павлов	Максим	6600	5.00
2	Сидоров	Алексей	7000	5.00
3	Андреев	Андрей	5500	5.00
4	Смирнов	Дмитрий	4000	4.50
5	Лебедев	Владимир	2500	4.50
6	Сергеев	Павел	4100	4.25
7	Зайцев	Илья	4200	4.00
8	Новиков	Михаил	2000	4.00
9	Кузьмин	Кирилл	3900	4.00
10	Горшков	Константин	3700	4.00
11	Кузнецова	Мария	3000	4.00
12	Васильевский	Денис	0	3.50
13	Кузнецов	Сергей	0	3.00
14	Новиков	Олег	0	3.00
15	Романов	Виктор	0	3.00
16	Морозов	Евгений	0	3.00
17	Козлов	Артем	0	3.00
18	Попов	Николай	0	3.00
19	Васильев	Роман	0	2.50

Рисунок 16 – Результат выполнения запроса

3.3 DELETE

- Удалить пересдачи студентов, если после пересдачи они получили всё равно двойку.

Сначала необходимо вывести студентов, которые получили двойку на 2/3 попытке. Для этого я написала запрос, представленный в листинге 16.

```

SELECT
    student.surname,
    student.name,
    subject.name AS subject_name,
    attestation.mark,
    attestation.attempt_number
FROM attestation
JOIN student ON student.id = attestation.student_id
JOIN subject ON subject.id = attestation.subject_id
WHERE attestation.mark = 2
      AND attestation.attempt_number > 1
ORDER BY student.surname, subject.name, attempt_number;

```

Листинг 16 – Просмотр студентов, не сдавших экзамен со 2–3 попытки

В результате выполнения запроса вывелась таблица, показанная на рисунке 17.

Data Output Сообщения Notifications					
Showing rows: 1 to 6					
	surname character varying (30)	name character varying (30)	subject_name character varying (100)	mark integer	attempt_number integer
1	Андреев	Андрей	Дискретная математика	2	3
2	Королев	Аркадий	Дискретная математика	2	2
3	Кузнецова	Мария	Математический анализ	2	3
4	Новиков	Олег	Линейная алгебра	2	3
5	Романов	Виктор	Компьютерные сети	2	3
6	Соболев	Александр	Компьютерные сети	2	3

Рисунок 17 – Результат выполнения запроса

Далее нужно удалить записи о неудачных попытках студентов. Мной был написан SQL-запрос, позволяющий сделать это. Он представлен в листинге 17.

```
DELETE FROM attestation
WHERE id IN (
  SELECT id
  FROM attestation
  WHERE attempt_number > 1
  AND mark = 2
);
```

Листинг 17 – Удаление записей о неудачных попытках пересдачи

Вновь прописав запрос, показанный в листинге 16, выводим таблицу студентов-двоечников (см. рис. 18). Она стала пустой.

Data Output Сообщения Notifications					
	surname character varying (30)	name character varying (30)	subject_name character varying (100)	mark integer	attempt_number integer

Рисунок 18 – Результат выполнения запроса

4. Создание индексов

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. В отчете отразить все команды, время запросов без индекса и с индексами.

4.1 Запрос №1

Сначала я прописала SQL-запрос с анализом для вывода информации об аттестации студентов, где идентификатор лежит в отрезке [1, 20]. Данные я взяла из таблицы attestation. Запрос представлен в листинге 18.

```
EXPLAIN ANALYZE
SELECT *
FROM attestation
WHERE student_id BETWEEN 1 AND 20;
```

Листинг 18 – Запрос с анализом

В результате выполнения запроса я получила таблицу, представленную на рисунке 19. Время выполнения составило 0.053 ms.

Data Output		Сообщения		Notifications	
<div><div>☰</div><div>📄</div><div>▼</div><div>📄</div><div>▼</div><div>🗑️</div><div>📄</div><div>⬇️</div><div>📈</div><div>SQL</div></div>		Showing rows: 1 to 5			
		QUERY PLAN			
		text			
1	Seq Scan on attestation (cost=0.00..19.60 rows=3 width=98) (actual time=0.026..0.031 rows=36 loops...				
2	Filter: ((student_id >= 1) AND (student_id <= 20))				
3	Rows Removed by Filter: 2				
4	Planning Time: 0.100 ms				
5	Execution Time: 0.053 ms				

Рисунок 19 – Результат выполнения запроса

Далее я создала индекс, запрос с созданием показан в листинге 19.

```
CREATE INDEX idx_attestation_student_id
ON attestation (student_id);
```

Листинг 19 – Создание индекса

После создания индекса снова прописываю запрос с анализом, показанный в листинге 18. Результат выполнения представлен на рисунке 20.

Data Output		Сообщения		Notifications	
<div><div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div></div><div>Showing rows: 1 to</div></div>					
QUERY PLAN					
text					
1	Seq Scan on attestation (cost=0.00..1.57 rows=1 width=98) (actual time=0.017..0.021 rows=36 loops...				
2	Filter: ((student_id >= 1) AND (student_id <= 20))				
3	Rows Removed by Filter: 2				
4	Planning Time: 2.336 ms				
5	Execution Time: 0.043 ms				

Рисунок 20 – Результат выполнения запроса

Время выполнения сократилось до 0.043 ms, что на 19% меньше исходного времени. Так как таблица attestation содержит не так много строк, нужно проверить на другой таблице.

4.2 Запрос №2

Я прописала такой же SQL-запрос с анализом, но данные я взяла из таблицы student. Запрос представлен в листинге 20.

```
EXPLAIN ANALYZE
SELECT *
FROM student
WHERE id BETWEEN 1 AND 20;
```

Листинг 20 – Запрос с анализом

В результате выполнения запроса получаем таблицу, представленную на рисунке 21. Время выполнения составило 0.117 ms.

Data Output		Сообщения	Notifications
<div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><</div></div></div>			

Рисунок 21 – Результат выполнения запроса

Далее необходимо создать индекс. Запрос с созданием показан в листинге 21.

```
CREATE INDEX idx_student_id ON student (id);
```

Листинг 21 – Создание индекса

После успешно созданного индекса повторяю запрос, указанный в листинге 20. Результат выполнения представлен на рисунке 22.

Data Output		Сообщения		Notifications	
<div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>		<div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>		<div>Showing rows: 1 t</div>	
		QUERY PLAN			
		text			
1	Seq Scan on student (cost=0.00..1.79 rows=19 width=57) (actual time=0.023..0.030 rows=20 loops...				
2	Filter: ((id >= 1) AND (id <= 20))				
3	Rows Removed by Filter: 33				
4	Planning Time: 3.526 ms				
5	Execution Time: 0.056 ms				

Рисунок 22 – Результат выполнения запроса

Время выполнения сократилось до 0.056 ms, что на 53% меньше исходного времени.

После работы с индексами их необходимо удалить. Я делаю это с помощью команд, представленных в листинге 22.

```
DROP INDEX idx_student_id;
DROP INDEX idx_attestation_student_id;
```

Листинг 22 – Удаление индексов

Выводы

В ходе работы были выполнены запросы на модификацию данных с использованием подзапросов: вставка, обновление и удаление записей. Проведено создание индексов для ускорения выборок и выполнено сравнение времени выполнения запросов с индексами и без них. Установлено, что индексация значительно повышает производительность при работе с большими объёмами данных. Работа позволила закрепить навыки оптимизации запросов и правильного проектирования структуры базы данных.