Министерство науки и высшего образования РоссийскойФедерации
федеральное государственное автономное образовательноеучреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL»

Студент: Злотникова Карина Александровна

Факультет: ИКТ

Группа: K32392

Преподаватель: Говорова М.М.

Дата: 02.05.2023

Санкт-Петербург 2023

Лабораторная №2

Цель работы:

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

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL(согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) сиспользованием подзапросов.
3. Изучить графическое представление запросов и посмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов исравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Индивидуальное практическое задание: «Учет выполнения заданий»

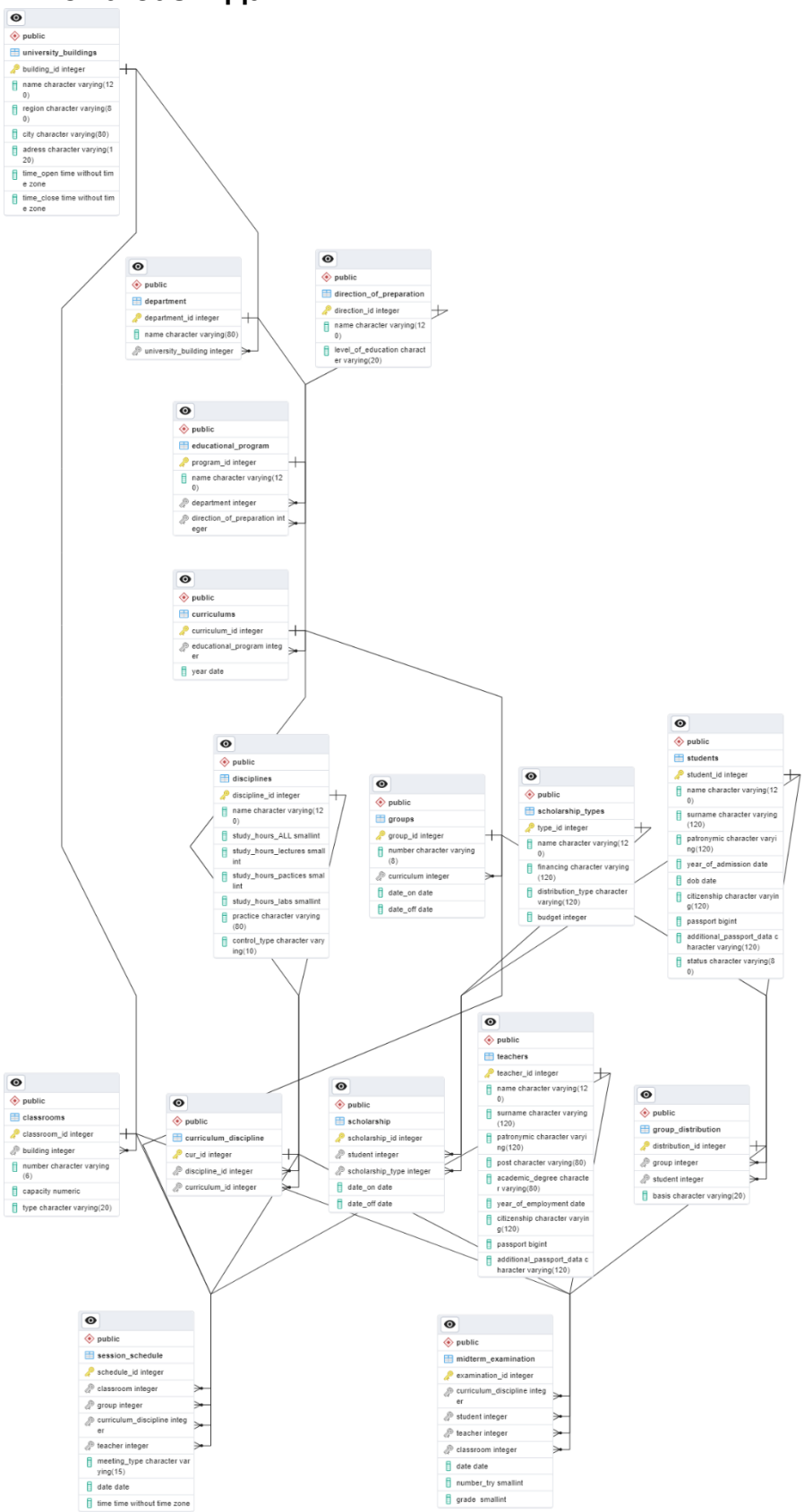
Запросы:

- 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 заданному направлению по результатам сдачи сессии, упорядочить 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.

Представления:

- 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).

Схема базы данных:



Запросы

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

```
SELECT DISTINCT d.name, s.date, t.surname
FROM group_distribution gd
JOIN students st ON gd.student = st.student_id
JOIN groups g ON gd.group = g.group_id
JOIN session_schedule s ON gd.group = s.group
JOIN curriculum_discipline cd ON s.curriculum_discipline = cd.discipline_id
JOIN disciplines d ON d.discipline_id = cd.discipline_id
JOIN teachers t ON s.teacher = t.teacher_id
WHERE g.group_id = 10;
```

	name character varying (120)	date date	surname character varying (120)
1	Матаматика	2023-05-21	Петров

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

```
SELECT DISTINCT st.surname, t.surname
FROM students st
JOIN group_distribution gd ON st.student_id = gd.student
JOIN midterm_examination s ON gd.student = s.student
JOIN teachers t ON s.teacher = t.teacher_id
WHERE s.grade <60 AND s.number_try = 1;
```

	surname character varying (120)	surname character varying (120)
1	Смирнов	Сидорова
2	Зайцева	Петров
3	Волкова	Петров
4	Федоров	Иванов
5	Ковалева	Иванов
6	Козлова	Козлов
7	Иванов	Иванов
8	Воробьев	Козлов
9	Максимова	Козлов
10	Калинин	Козлов
11	Зайцева	Сидорова
12	Морозова	Петров
13	Егорова	Иванов
14	Михайлова	Сидорова
15	Козлов	Козлов
16	Козлов	Иванов

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

```
SELECT DISTINCT st.surname
FROM students st
JOIN group_distribution gd ON st.student_id = gd.student
JOIN midterm_examination s ON gd.student = s.student
WHERE s.grade > (SELECT AVG(grade) FROM midterm_examination WHERE curriculum_discipline
= 1);
```

	surname character varying (120)
1	Сидорова
2	Васильева
3	Иванов
4	Новикова
5	Федоров
6	Максимова
7	Морозова
8	Ильин
9	Петров
10	Михайлова
11	Козлова
12	Воробьев
13	Зайцева
14	Смирнов
15	Ковалева
16	Лебедев

- Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убыванию:

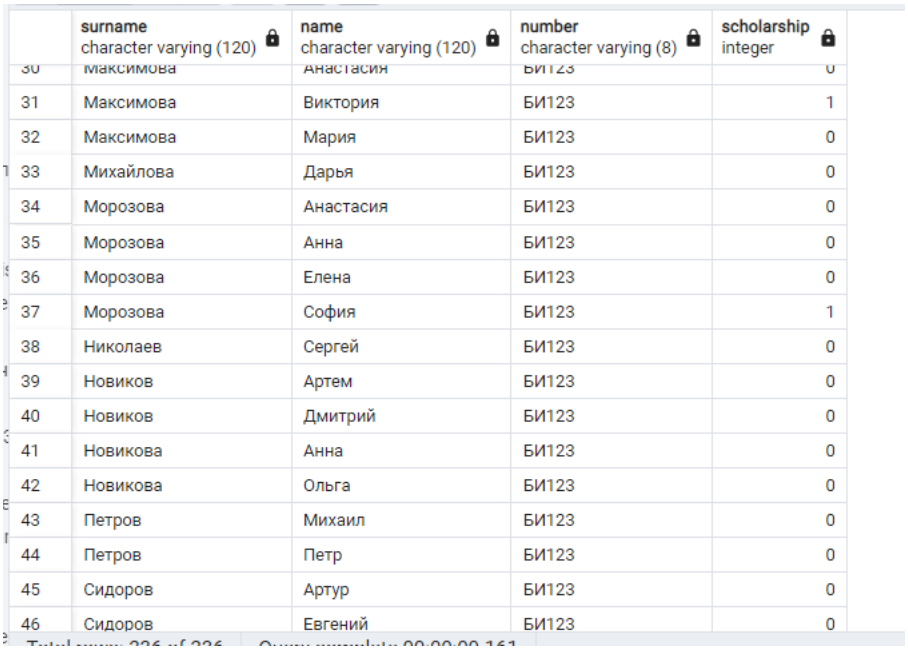
```
SELECT g.number, AVG(s.grade) AS average_grade
FROM groups g
JOIN group_distribution gd ON g.group_id = gd.group
JOIN students st ON st.student_id = gd.student
JOIN midterm_examination s ON gd.student = s.student
JOIN curriculum_discipline cd ON s.curriculum_discipline = cd.discipline_id

JOIN curriculums c ON c.curriculum_id = cd.curriculum_id
JOIN educational_program ep ON ep.program_id = c.educational_program
JOIN direction_of_preparation dp ON dp.direction_id = ep.direction_of_preparation
WHERE dp.name = 'Мвыавав'
GROUP BY g.number
ORDER BY average_grade DESC;
```

	number character varying (8)	average_grade numeric
1	ИУ522	74.9814814814814815
2	БИ123	73.8767908309455587
3	ПМ420	73.7363896848137536
4	ИУ721	73.6068376068376068

- 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.

```
SELECT s.surname, s.name, g.number,
 CASE
  WHEN COUNT(me.student) = COUNT(CASE WHEN me.grade >= 75 THEN me.student END)
THEN 1
  ELSE 0
 END AS scholarship
FROM students s
JOIN group_distribution gd ON s.student_id = gd.student
JOIN midterm_examination me ON gd.student = me.student
JOIN groups g ON gd.group = g.group_id
GROUP BY s.surname, s.name, g.number
ORDER BY g.number, s.surname;
```



	surname character varying (120)	name character varying (120)	number character varying (8)	scholarship integer
30	Максимова	Анастасия	БИ123	0
31	Максимова	Виктория	БИ123	1
32	Максимова	Мария	БИ123	0
33	Михайлова	Дарья	БИ123	0
34	Морозова	Анастасия	БИ123	0
35	Морозова	Анна	БИ123	0
36	Морозова	Елена	БИ123	0
37	Морозова	София	БИ123	1
38	Николаев	Сергей	БИ123	0
39	Новиков	Артем	БИ123	0
40	Новиков	Дмитрий	БИ123	0
41	Новикова	Анна	БИ123	0
42	Новикова	Ольга	БИ123	0
43	Петров	Михаил	БИ123	0
44	Петров	Петр	БИ123	0
45	Сидоров	Артур	БИ123	0
46	Сидоров	Евгений	БИ123	0

- Вывести список студентов, сдавших все положенные экзамены.

```
SELECT
   s.surname,
   s.name
FROM
   students s
   LEFT JOIN midterm_examination me ON s.student_id = me.student
GROUP BY
   s.surname,
   s.name
HAVING
   COUNT(me.examination_id) = COUNT(CASE WHEN me.grade > 60 THEN 1 END)
   AND COUNT(me.examination_id) > 0;
```

	surname character varying (120)	name character varying (120)
1	Морозова	София
2	Федоров	Илья
3	Морозова	Анна
4	Волков	Александр
5	Петров	Михаил
6	Максимова	Мария
7	Федоров	Георгий
8	Волкова	Анастасия
9	Сидорова	Анна
10	Зайцева	Светлана
11	Кузьмин	Игорь
12	Новиков	Артем
13	Волкова	Мария
14	Ильин	Алексей
15	Козлов	Иван
16	Соколов	Иван

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

```
SELECT
    g.number AS group_number,
    s.surname,
    AVG(me.grade) AS average_grade
FROM
    groups g
    INNER JOIN group_distribution gd ON g.group_id = gd.group
    INNER JOIN students s ON gd.student = s.student_id
    INNER JOIN midterm_examination me ON s.student_id = me.student
GROUP BY
    g.number,
    s.surname
HAVING
    AVG(me.grade) = (
        SELECT
            MAX(avg_grade)
        FROM (
            SELECT
                g.number AS group_number,
                AVG(me.grade) AS avg_grade
            FROM
                groups g
                INNER JOIN group_distribution gd ON g.group_id =
gd.group
                INNER JOIN students s ON gd.student = s.student_id
                INNER JOIN midterm_examination me ON s.student_id =
me.student
            GROUP BY
                g.number
        ) AS subquery
        WHERE
            g.number = subquery.group_number
    );
```

	group_number character varying (8)	surname character varying (120)	average_grade numeric
1	ПМ420	Лебедев	91.0000000000000000
2	ИУ721	Егорова	79.0000000000000000
3	БИ123	Федоров	87.0000000000000000

- UPDATE с использованием подзапроса: Отчисление студента, если он получил меньше 60 баллов на третьей попытке экзамена

```
UPDATE public.students
SET status = 'отчислен'
WHERE student_id IN (
    SELECT student
    FROM public.midterm_examination
    WHERE number_try = 3
    GROUP BY student
    HAVING MIN(grade) < 60
);


--(список студентов на отчисление)

SELECT *
FROM public.students s
WHERE EXISTS (
  SELECT *
  FROM public.midterm_examination me
  WHERE me.student = s.student_id
  AND me.number_try = 3
  AND me.grade < 60
);
```

	student_id [PK] integer	name character varying (120)	surname character varying (120)	patronymic character varying (120)	year_of_admission date	dob date	citizenship character varying (120)	passport bigint	additional_passport_data character varying (120)	status character varying (80)
1	136	Анастасия	Морозова	Алексеевна	2021-09-01	2000-07-24	Россия	1357913573	[null]	студент
2	92	Артем	Козлов	Артемович	2021-09-01	2000-11-11	Россия	1357913579	[null]	студент
3	97	Елена	Воробьева	Александровна	2021-09-01	2000-04-16	Россия	5432167890	[null]	студент
4	104	Денис	Смирнов	Денисович	2021-09-01	2000-11-23	Россия	9876543212	[null]	студент
5	109	Владимир	Калинин	Дмитриевич	2021-09-01	2000-04-28	Россия	2468135793	[null]	студент
6	111	Иван	Сидоров	Иванович	2021-09-01	2000-06-30	Россия	1234567892	[null]	студент
7	113	Никита	Федоров	Никитич	2021-09-01	2000-08-01	Россия	4321098767	[null]	студент
8	117	Мария	Козлова	Максимовна	2021-09-01	2000-12-05	Россия	8642097533	[null]	студент
9	119	Анастасия	Воробьева	Александровна	2021-09-01	2000-02-07	Россия	5432167892	[null]	студент
10	121	Алиса	Максимова	Дмитриевна	2021-09-01	2000-04-09	Россия	9876543215	[null]	студент

	student_id [PK] integer	name character varying (120)	surname character varying (120)	patronymic character varying (120)	year_of_admission date	dob date	citizenship character varying (120)	passport bigint	additional_passport_data character varying (120)	status character varying (80)
1	136	Анастасия	Морозова	Алексеевна	2021-09-01	2000-07-24	Россия	1357913573	[null]	отчислен
2	92	Артем	Козлов	Артемович	2021-09-01	2000-11-11	Россия	1357913579	[null]	отчислен
3	97	Елена	Воробьева	Александровна	2021-09-01	2000-04-16	Россия	5432167890	[null]	отчислен
4	104	Денис	Смирнов	Денисович	2021-09-01	2000-11-23	Россия	9876543212	[null]	отчислен
5	109	Владимир	Калинин	Дмитриевич	2021-09-01	2000-04-28	Россия	2468135793	[null]	отчислен
6	111	Иван	Сидоров	Иванович	2021-09-01	2000-06-30	Россия	1234567892	[null]	отчислен
7	113	Никита	Федоров	Никитич	2021-09-01	2000-08-01	Россия	4321098767	[null]	отчислен
8	117	Мария	Козлова	Максимовна	2021-09-01	2000-12-05	Россия	8642097533	[null]	отчислен
9	119	Анастасия	Воробьева	Александровна	2021-09-01	2000-02-07	Россия	5432167892	[null]	отчислен
10	121	Алиса	Максимова	Дмитриевна	2021-09-01	2000-04-09	Россия	9876543215	[null]	отчислен

- `INSERT` с использованием подзапроса: Если у студента оценка больше 60, то ему не будет назначена новая попытка, а если оценка меньше 60, то будет добавлена новая запись с номером попытки на 1 больше.

```
INSERT INTO public.midterm_examination (curriculum_discipline,
  student, teacher, classroom, date, number_try, grade)
SELECT cd.cur_id, s.student_id, t.teacher_id, c.classroom_id,
  CURRENT_DATE, CASE WHEN e.number_try IS NULL THEN 1 ELSE
  LEAST(e.number_try + 1, 3) END, CASE WHEN e.grade < 60 THEN NULL
  ELSE e.grade END
FROM public.students AS s
JOIN public.group_distribution AS gd ON s.student_id = gd.student
JOIN public.curriculum_discipline AS cd ON gd.group = cd.curriculum_id
JOIN public.teachers AS t ON cd.discipline_id = t.teacher_id
JOIN public.classrooms AS c ON t.teacher_id = c.classroom_id
LEFT JOIN public.midterm_examination AS e ON cd.cur_id =
  e.curriculum_discipline AND s.student_id = e.student
  GROUP BY cd.cur_id, s.student_id, t.teacher_id, c.classroom_id,
  e.grade;
```

	examination_id [PK] integer	curriculum_discipline integer	student integer	teacher integer	classroom integer	date date	number_try smallint	grade smallint
22	3343	1	86	8	1	2023-05-20	1	92
23	3344	1	87	8	1	2023-05-20	1	79
24	3345	1	88	8	1	2023-05-20	1	94
25	3346	1	89	8	1	2023-05-20	1	60
26	3347	1	90	8	1	2023-05-20	1	78
27	3348	1	91	8	1	2023-05-20	1	95
28	3349	1	93	8	1	2023-05-20	1	50
29	3350	1	94	8	1	2023-05-20	1	64
30	3351	1	95	8	1	2023-05-20	1	77
31	3352	1	96	8	1	2023-05-20	1	76
32	3353	1	98	8	1	2023-05-20	1	90
33	3354	1	99	8	1	2023-05-20	1	88
34	3355	1	100	8	1	2023-05-20	1	67
35	3356	1	101	8	1	2023-05-20	1	89
36	3357	1	102	8	1	2023-05-20	1	61
37	3358	1	103	8	1	2023-05-20	1	67

	examination_id [PK] integer	curriculum_discipline integer	student integer	teacher integer	classroom integer	date date	number_try smallint	grade smallint
231	3940	1	93	8	1	2023-05-20	2	66
232	3941	1	118	8	1	2023-05-20	2	58
233	3942	1	125	8	1	2023-05-20	2	61
234	3943	1	130	8	1	2023-05-20	2	69
235	3944	1	133	8	1	2023-05-20	2	81
236	3945	1	134	8	1	2023-05-20	2	84
237	3946	1	137	8	1	2023-05-20	2	90
238	3947	2	1	9	2	2023-05-21	2	98
239	3948	2	3	9	2	2023-05-21	2	87
240	3949	2	4	9	2	2023-05-21	2	56
241	3950	2	7	9	2	2023-05-21	2	92
242	3951	2	8	9	2	2023-05-21	2	93
243	3952	2	86	9	2	2023-05-21	2	69
244	3953	2	89	9	2	2023-05-21	2	92
245	3954	2	90	9	2	2023-05-21	2	93
246	3955	2	91	9	2	2023-05-21	2	82

- DELETE с использованием подзапроса:
Удаление более поздних дубликатов таблицы студентов

```
DELETE FROM public.students
WHERE student_id NOT IN (
    SELECT min_id
    FROM (
        SELECT MIN(student_id) AS min_id
        FROM public.students
        GROUP BY name, surname, patronymic, year_of_admission, dob,
citizenship, passport, additional_passport_data, status
        HAVING COUNT(*) > 1
    ) AS duplicates

);
```

Создание представлений

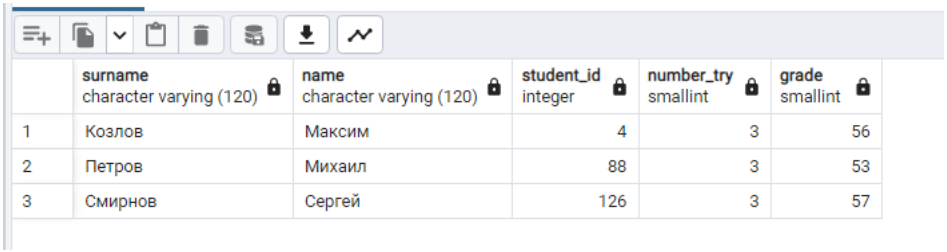
- список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;

```
CREATE VIEW public.students_with_retake_teacher AS
SELECT s.surname AS student_surname, t.surname AS teacher_surname
FROM public.students AS s
JOIN public.midterm_examination AS e ON s.student_id = e.student
JOIN public.teachers AS t ON e.teacher = t.teacher_id
WHERE e.number_try = 1 AND e.grade < 60;
```

	student_surname character varying (120)	teacher_surname character varying (120)
1	Козлов	Иванов
2	Ковалева	Иванов
3	Иванов	Иванов
4	Егорова	Иванов
5	Калинин	Иванов
6	Смирнов	Иванов
7	Козлов	Иванов
8	Калинин	Иванов
9	Федоров	Иванов
10	Максимова	Иванов
11	Морозова	Иванов
12	Сидоров	Иванов
13	Зайцева	Иванов
14	Морозова	Иванов
15	Козлов	Петров
16	Ковалева	Петров

- Представление данных о студентах при получении ими хотя бы одной оценки меньше 60 (после и на 3-й попытке).

```
CREATE VIEW students_with_low_grades AS
SELECT s.surname, s.name, s.student_id, me.number_try, me.grade
FROM students s
JOIN midterm_examination me ON s.student_id = me.student
  WHERE me.grade < 60 AND me.number_try >= 3;
```



	surname character varying (120)	name character varying (120)	student_id integer	number_try smallint	grade smallint
1	Козлов	Максим	4	3	56
2	Петров	Михаил	88	3	53
3	Смирнов	Сергей	126	3	57

Индексы

1) Запрос

```
EXPLAIN ANALYZE
SELECT s.student_id, s.name, s.surname, AVG(m.grade) AS average_grade
FROM students s
JOIN midterm_examination m ON s.student_id = m.student
GROUP BY s.student_id, s.name, s.surname
HAVING AVG(m.grade) > 70;
```

"Planning Time: 0.322 ms"
"Execution Time: 0.233 ms"

Создание индекса

CREATE INDEX idx_midterm_examination_student_id ON midterm_examination (student_id);

После создание индекса

"Planning Time: 0.208 ms"
"Execution Time: 0.206 ms"

2)Запрос

```
EXPLAIN ANALYZE
SELECT gd.group, COUNT(*) AS total_passing_students
FROM group_distribution gd
JOIN midterm_examination m ON gd.student = m.student

WHERE m.grade >= 60
GROUP BY gd.group;
```

"Planning Time: 0.744 ms"
"Execution Time: 0.367 ms"

Создание индекса

CREATE INDEX idx_midterm_examination_distribution_grade ON midterm_examination (student, grade);

После создание индекса

"Planning Time: 0.442 ms"
"Execution Time: 0.302 ms"

Выводы

В процессе выполнения лабораторной работы было освоено составление запросов INSERT, UPDATE и DELETE,а также изучено графическое представление запросов. Кроме того, были созданы индексы, что привело к уменьшению количества этапов при выполнении запросов и ускорению их выполнения.