

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)**

**Факультет прикладной информатики**

**Образовательная программа Мобильные и сетевые технологии**

**Направление подготовки 09.03.03 Мобильные и сетевые технологии**

**О Т Ч Е Т**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**"ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ.  
ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ"**

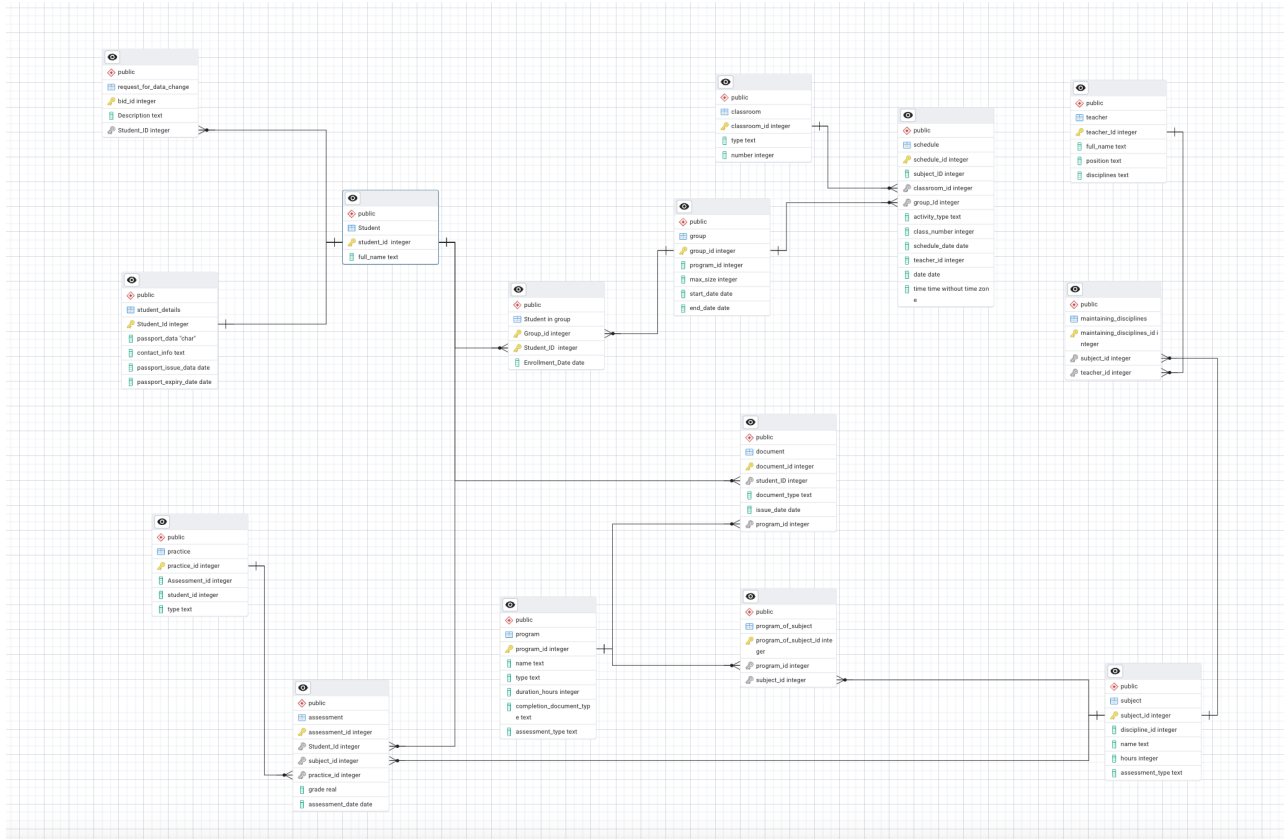
**Обучающийся:** Мельникова Настасья К3241

**Преподаватель:** Говорова Марина Михайловна

Санкт-Петербург  
2025

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

## 2. Схема базы данных (ЛР 3).



## 3. Выполнение:

### 3.1 Запросы к базе данных.

В рамках выполнения лабораторной работы я составила и выполнила SQL-запросы в соответствии с индивидуальным заданием (часть 2). Каждый запрос формировался исходя из требований задания, отражающих конкретные задачи по выборке данных из базы.

Запрос 1: Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей.

Запрос
История запросов

---

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

```

SELECT
    g.group_id,
    subj.name AS subject_name,
    sch.schedule_date,
    t.full_name AS teacher_name
FROM
    "group" g
JOIN
    program p ON g.program_id = p.program_id
JOIN
    program_of_subject pos ON pos.program_id = p.program_id
JOIN
    subject subj ON subj.subject_id = pos.subject_id
JOIN
    schedule sch ON sch."subject_ID" = subj.subject_id
JOIN
    teacher t ON sch."teacher_id" = t."teacher_Id"
WHERE
    g.group_id = 201
ORDER BY
    sch.schedule_date;

```

---

Data Output
Сообщения
Notifications

---

	group_id integer	subject_name text	schedule_date date	teacher_name text
1	201	Математика	2024-09-02	Смирнов Алексей Петрович
2	201	Информатика	2024-09-02	Иванова Ольга Сергеевна

Запрос 2: Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```

1  SELECT
2      st.full_name AS student_name,
3      subj.name AS subject_name,
4      a.grade,
5      t.full_name AS teacher_name
6  FROM
7      assessment a
8  JOIN
9      "Student" st ON st."student_id " = a."Student_Id"
10 JOIN
11     subject subj ON subj.subject_id = a.subject_id
12 JOIN
13     maintaining_disciplines md ON md.subject_id = subj.subject_id
14 JOIN
15     teacher t ON t."teacher_Id" = md.teacher_id
16 WHERE
17     a.grade = 2
18     AND a.assessment_date = (
19         SELECT MIN(a2.assessment_date)
20         FROM assessment a2
21         WHERE a2."Student_Id" = a."Student_Id"
22             AND a2.subject_id = a.subject_id
23     )
24 ORDER BY
25     st.full_name;
26

```

Data Output Сообщения Notifications

	student_name text	subject_name text	grade real	teacher_name text
1	Кузнецов Дмитрий Алексеевич	Физика	2	Петров Дмитрий Викторович
2	Алексеева Мария Ивановна	Математика	2	Смирнов Алексей Петрович

Запрос 3: Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине.

ЗапросИстория запросов

1

SELECT

2

st.full\_name AS student\_name,

3

subj.name AS subject\_name,

4

a.grade,

5

avg\_grades.avg\_grade

6

FROM

7

assessment a

8

JOIN

9

"Student" st ON st."student\_id " = a."Student\_Id"

10

JOIN

11

subject subj ON subj.subject\_id = a.subject\_id

12

JOIN (

13

SELECT

14

subject\_id,

15

AVG(grade) AS avg\_grade

16

FROM

17

assessment

18

GROUP BY

19

subject\_id

20

) avg\_grades ON avg\_grades.subject\_id = a.subject\_id

21

WHERE

22

a.grade > avg\_grades.avg\_grade

23

ORDER BY

24

subj.name, a.grade DESC;

25

26

Data OutputСообщенияNotifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	student_name text	subject_name text	grade real	avg_grade double precision
1	Алексеева Мария Ивановна	Математика	4	3

Запрос 4: Создать рейтинговый список групп по успеваемости по заданному направлению по результатам сдачи сессии, упорядочив его по убыванию.

```

1  SELECT
2      g.group_id,
3      p.name AS program_name,
4      ROUND(AVG(a.grade)::numeric, 2) AS average_grade
5  FROM
6      assessment a
7  JOIN
8      "Student" s ON s."student_id " = a."Student_Id"
9  JOIN
10     "Student in group" sg ON sg."Student_ID " = s."student_id "
11  JOIN
12     "group" g ON g.group_id = sg."Group_id"
13  JOIN
14     program p ON p.program_id = g.program_id
15  WHERE
16     p.program_id = 1
17  GROUP BY
18     g.group_id, p.name
19  ORDER BY
20     ROUND(AVG(a.grade)::numeric, 2) DESC;
21
22
23

```

	group_id integer	program_name text	average_grade numeric
1	201	Программа по математике	3.00
2	202	Программа по математике	3.00

Запрос 5: 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.

```

1  SELECT
2      g.group_id,
3      s.full_name AS student_name,
4      ROUND(AVG(a.grade)::numeric, 2) AS average_grade,
5      CASE
6          WHEN SUM(CASE WHEN a.grade = 3 THEN 1 ELSE 0 END) = 0 THEN 1
7          ELSE 0
8      END AS scholarship
9  FROM
10     assessment a
11  JOIN
12     "Student" s ON s."student_id" = a."Student_Id"
13  JOIN
14     "Student in group" sg ON sg."Student_ID" = s."student_id "
15  JOIN
16     "group" g ON g.group_id = sg."Group_id"
17  GROUP BY
18     g.group_id, s.full_name
19  ORDER BY
20     g.group_id, s.full_name;
21
22
23

```

Data Output Сообщения Notifications

	group_id integer	student_name text	average_grade numeric	scholarship integer
1	201	Алексеева Мария Иванов...	3.00	1
2	202	Иванов Пётр Сергеевич	3.00	0
3	203	Сидорова Анна Викторовна	4.00	1

Запрос 6: Вывести список студентов, сдавших все положенные экзамены.

```

1  SELECT
2      s.full_name AS student_name
3  FROM
4      "Student" s
5  JOIN
6      "Student in group" sg ON sg."Student_ID " = s."student_id "
7  JOIN
8      "group" g ON g.group_id = sg."Group_id"
9  JOIN
10     program p ON p.program_id = g.program_id
11  JOIN
12     program_of_subject pos ON pos.program_id = p.program_id
13  -- проверка всех дисциплин, положенных студенту
14  WHERE NOT EXISTS (
15      SELECT 1
16      FROM program_of_subject required
17      WHERE required.program_id = p.program_id
18          AND NOT EXISTS (
19              SELECT 1
20              FROM assessment a
21              WHERE a."Student_Id" = s."student_id "
22                  AND a.subject_id = required.subject_id
23                  AND a.grade >= 3
24          )
25  )
26  ORDER BY
27      s.full_name;

```



	student_name text	
1	Сидорова Анна Викторовна	



Запрос 7: Вывести список студентов, получивших максимальный средний балл в своей группе.

```
WITH student_avg AS (  
    SELECT  
        s."student_id ",  
        s.full_name,  
        sg."Group_id",  
        ROUND(AVG(a.grade)::numeric, 2) AS avg_grade  
    FROM  
        "Student" s  
    JOIN  
        "Student in group" sg ON sg."Student_ID " = s."student_id "  
    JOIN  
        assessment a ON a."Student_Id" = s."student_id "  
    GROUP BY  
        s."student_id ", s.full_name, sg."Group_id"  
) ,  
  
group_max_avg AS (  
    SELECT  
        "Group_id",  
        MAX(avg_grade) AS max_avg  
    FROM  
        student_avg  
    GROUP BY  
        "Group_id"  
)  
  
SELECT  
    sa.full_name,  
    sa."Group_id",  
    sa.avg_grade  
FROM  
    student_avg sa  
JOIN  
    group_max_avg gm  
    ON sa."Group_id" = gm."Group_id" AND sa.avg_grade = gm.max_avg  
ORDER BY  
    sa."Group_id", sa.full_name;
```

Запрос

История запросов

---

13

GROUP BY

14

s."student\_id ", s.full\_name, sg."Group\_id"

15

),

16

17

group\_max\_avg AS (

18

SELECT

19

"Group\_id",

20

MAX(avg\_grade) AS max\_avg

21

FROM

---

Data Output

Сообщения

Notifications

---

≡+

▼

▼

SQL

---

	full_name text	Group_id integer	avg_grade numeric
1	Тимофеев Андрей Олегович	101	4.67
2	Васильев Николай Артёмович	102	5.00
3	Ковалёва Ирина Михайловна	103	3.00
4	Алексеева Мария Ивановна	201	4.00
5	Иванов Пётр Сергеевич	202	3.00
6	Сидорова Анна Викторовна	203	4.50

## 3.2 Представления

Представление 1: список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен;

```

CREATE OR REPLACE VIEW vw_students_first_try_fail AS
SELECT
    s.full_name AS student_name,
    subj.name AS subject_name,
    a.grade,
    t.full_name AS teacher_name
FROM
    assessment a
JOIN
    "Student" s ON s."student_id " = a."Student_Id"
JOIN
    subject subj ON subj.subject_id = a.subject_id
JOIN
    maintaining_disciplines md ON md.subject_id = subj.subject_id
JOIN
    teacher t ON t."teacher_Id" = md.teacher_id
WHERE
    a.grade = 2
    AND a.assessment_date = (
        SELECT MIN(a2.assessment_date)
        FROM assessment a2
        WHERE a2."Student_Id" = a."Student_Id"
        AND a2.subject_id = a.subject_id
    );

```

**Представление 2: Студенты, получившие 2 после третьей попытки**

---

```
CREATE OR REPLACE VIEW vw_students_fail_after_3rd_attempt AS
SELECT
    s.full_name AS student_name,
    subj.name AS subject_name,
    a.grade,
    a.assessment_date
FROM
    assessment a
JOIN
    "Student" s ON s."student_id " = a."Student_Id"
JOIN
    subject subj ON subj.subject_id = a.subject_id
WHERE
    a.grade = 2
    AND (
        SELECT COUNT(*)
        FROM assessment a2
        WHERE a2."Student_Id" = a."Student_Id"
            AND a2.subject_id = a.subject_id
            AND a2.assessment_date <= a.assessment_date
    ) >= 3;
```

### 3.3 Запросы на модификацию данных

Выполнение запросов на модификацию данных (INSERT, UPDATE, DELETE) с подзапросами INSERT

## Запрос История запросов

1	▼	INSERT INTO "Student" ("student_id ", full_name)
2		VALUES (
3		(SELECT MAX("student_id ") + 1 FROM "Student"),
4		'Петров Петр Петрович'
5		);
6		
7		
8		
9		
10		

## Data Output Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 95 msec.

до

Запрос

История запросов

1

2

3

4

SELECT

\*

FROM

"Student"

Data Output

Сообщения

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	student_id [PK] integer	full_name text
1	1	Алексеева Мария Ивановна
2	2	Иванов Пётр Сергеевич
3	3	Сидорова Анна Викторовна
4	4	Кузнецов Дмитрий Алексеевич
5	5	Новикова Елена Павловна
6	6	Тимофеев Андрей Олегович
7	7	Смирнова Виктория Юрьевна
8	8	Васильев Николай Артёмович
9	9	Ковалёва Ирина Михайловна
10	10	Григорьев Артём Дмитриевич
11	11	Фёдорова Ольга Константиновна
12	12	Морозов Алексей Владимирович
13	13	Михеева Татьяна Сергеевна
14	14	Яковлев Иван Владимирович
15	15	Павлова Екатерина Романовна

ПОСЛЕ

Запрос

История запросов

1

2

3

SELECT \* FROM "Student"

Data Output

Сообщения

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	student_id [PK] integer	full_name text
1	1	Алексеева Мария Ивановна
2	2	Иванов Пётр Сергеевич
3	3	Сидорова Анна Викторовна
4	4	Кузнецов Дмитрий Алексеевич
5	5	Новикова Елена Павловна
6	6	Тимофеев Андрей Олегович
7	7	Смирнова Виктория Юрьевна
8	8	Васильев Николай Артёмович
9	9	Ковалёва Ирина Михайловна
10	10	Григорьев Артём Дмитриевич
11	11	Фёдорова Ольга Константиновна
12	12	Морозов Алексей Владимирович
13	13	Михеева Татьяна Сергеевна
14	14	Яковлев Иван Владимирович
15	15	Павлова Екатерина Романовна
16	16	Петров Петр Петрович

Запрос UPDATE с подзапросом:

```

UPDATE assessment
SET grade = LEAST(grade + 1, 5)
WHERE "Student_Id" IN (
    SELECT "Student_Id"
    FROM assessment
    GROUP BY "Student_Id"
    HAVING AVG(grade) < 3.5
);

```

ДО

	assessment_id [PK] integer	Student_Id integer	subject_id integer	practice_id integer	grade real	assessment_date date
1	1	1	101	1001	2	2025-05-20
2	2	2	101	1001	3	2025-05-20
3	3	3	102	1001	4	2025-05-22
4	4	1	101	1002	4	2025-05-27
5	5	4	103	1001	2	2025-05-25

## ПОСЛЕ

Запрос

История запросов

1

2

SELECT \* FROM "assessment"

Data Output

Сообщения

Notifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

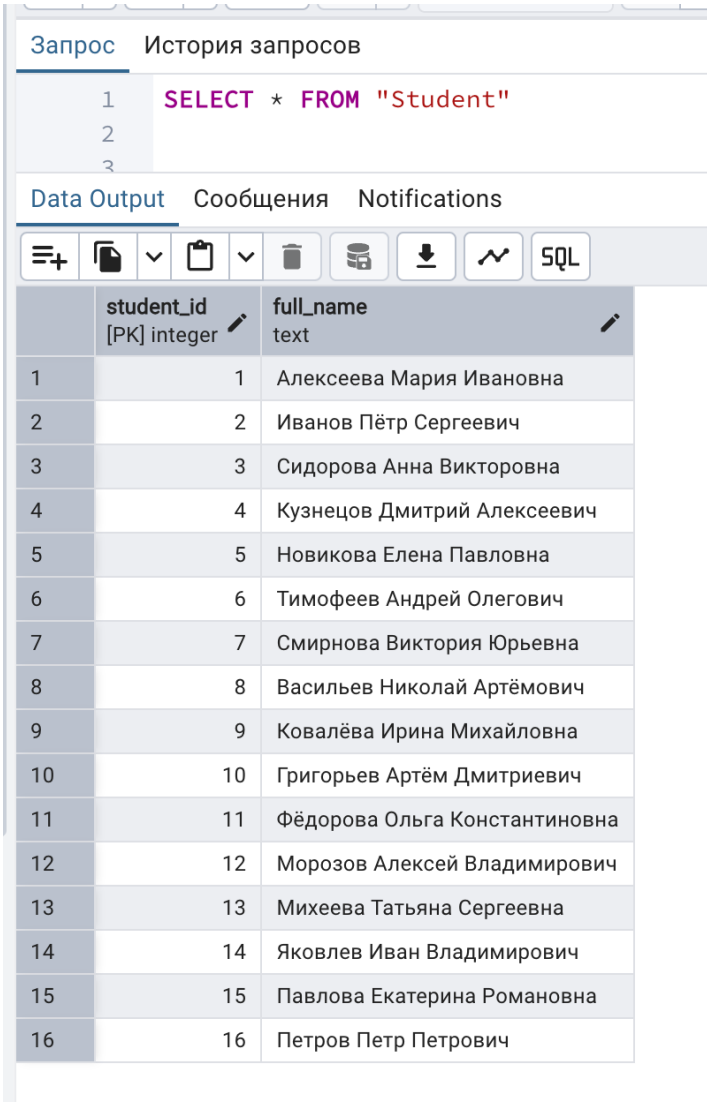
	assessment_id [PK] integer	Student_Id integer	subject_id integer	practice_id integer	grade real	assessment_date date
1	6	1	101	1001	5	2025-05-15
2	7	1	107	1001	4	2025-05-17
3	8	1	108	1001	5	2025-05-19
4	1	1	101	1001	2	2025-05-20
5	3	3	102	1001	4	2025-05-22
6	4	1	101	1002	4	2025-05-27
7	12	6	101	1001	5	2025-05-15
8	13	6	107	1001	4	2025-05-17
9	14	6	108	1001	5	2025-05-19
10	15	7	101	1001	4	2025-05-15
11	16	7	107	1001	3	2025-05-17
12	17	7	108	1001	4	2025-05-19
13	18	3	103	1001	5	2025-05-15
14	19	3	104	1001	4	2025-05-17
15	20	3	109	1001	5	2025-05-19
16	24	8	103	1001	5	2025-05-15
17	25	8	104	1001	5	2025-05-17
18	26	8	109	1001	5	2025-05-19
19	27	5	101	1001	4	2025-05-15
20	28	5	103	1001	5	2025-05-17
21	29	5	105	1001	5	2025-05-19

Запрос DELETE с подзапросом:



```
DELETE FROM "Student"
WHERE "student_id " = (
    SELECT MAX("student_id ") FROM "Student"
);
```

ДО



The screenshot shows a database management interface. At the top, there's a tab labeled 'Запрос' (Query) with a sub-tab 'История запросов' (Query History). Below this, a list of queries is shown, with the first one selected: 'SELECT \* FROM "Student"'. Below the query list, there are tabs for 'Data Output', 'Сообщения' (Messages), and 'Notifications'. The 'Data Output' tab is active, showing a table of results. The table has two columns: 'student\_id [PK] integer' and 'full\_name text'. The results are listed in 16 rows, numbered 1 to 16. Each row contains a student's ID and their full name.

	student_id [PK] integer	full_name text
1	1	Алексеева Мария Ивановна
2	2	Иванов Пётр Сергеевич
3	3	Сидорова Анна Викторовна
4	4	Кузнецов Дмитрий Алексеевич
5	5	Новикова Елена Павловна
6	6	Тимофеев Андрей Олегович
7	7	Смирнова Виктория Юрьевна
8	8	Васильев Николай Артёмович
9	9	Ковалёва Ирина Михайловна
10	10	Григорьев Артём Дмитриевич
11	11	Фёдорова Ольга Константиновна
12	12	Морозов Алексей Владимирович
13	13	Михеева Татьяна Сергеевна
14	14	Яковлев Иван Владимирович
15	15	Павлова Екатерина Романовна
16	16	Петров Петр Петрович

ПОСЛЕ

Запрос

История запросов

1

2

3

4

SELECT \* FROM "Student"

Data Output

Сообщения

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	student_id [PK] integer	full_name text
1	1	Алексеева Мария Ивановна
2	2	Иванов Пётр Сергеевич
3	3	Сидорова Анна Викторовна
4	4	Кузнецов Дмитрий Алексеевич
5	5	Новикова Елена Павловна
6	6	Тимофеев Андрей Олегович
7	7	Смирнова Виктория Юрьевна
8	8	Васильев Николай Артёмович
9	9	Ковалёва Ирина Михайловна
10	10	Григорьев Артём Дмитриевич
11	11	Фёдорова Ольга Константиновна
12	12	Морозов Алексей Владимирович
13	13	Михеева Татьяна Сергеевна
14	14	Яковлев Иван Владимирович
15	15	Павлова Екатерина Романовна

### 3.4 Создание индексов

В данном пункте лабораторной работы я выполнила следующие действия:

#### 1. Выполнение тестовых запросов без индексов

Были выполнены два выбранных запроса к базе данных, и с помощью команды EXPLAIN ANALYZE получены планы их выполнения.

Зафиксировано время выполнения и использованные методы доступа к данным (например, последовательное сканирование).

Выберем 2 запроса:

## Запрос №1

Запрос

История запросов

1

2

3

4

5

6

7

EXPLAIN ANALYZE

SELECT \*

FROM assessment

WHERE subject\_id = 101;

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

QUERY PLAN

text

🔒

1

2

3

4

5

Seq Scan on assessment (cost=0.00..1.06 rows=1 width=24) (actual time=0.034..0.038 rows=9 loops=...

Filter: (subject\_id = 101)

Rows Removed by Filter: 25

Planning Time: 0.067 ms

Execution Time: 0.052 ms

## Запрос №2:

Запрос

История запросов

1

2

3

4

5

6

7

EXPLAIN ANALYZE

SELECT \*

FROM assessment

WHERE subject\_id = 101 AND grade = 5;

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

QUERY PLAN

text

🔒

1

2

3

4

5

Seq Scan on assessment (cost=0.00..1.51 rows=1 width=24) (actual time=0.020..0.029 rows=2 loops=...

Filter: ((subject\_id = 101) AND (grade = '5':double precision))

Rows Removed by Filter: 32

Planning Time: 0.129 ms

Execution Time: 0.047 ms

Execute script

F5

## 2. Создание индексов

Для каждого из запросов были созданы:  
простой индекс (по одному полю),

Запрос

История запросов

1

2

3

4

5

6

7

▼

```
CREATE INDEX idx_assessment_subject_id
ON assessment(subject_id);
```

Data Output

Сообщения

Notifications

```
CREATE INDEX
```

Запрос завершён успешно, время выполнения: 105 msec.

составной индекс (по нескольким полям), с целью оптимизации скорости выборки данных

Запрос

История запросов

1

2

3

4

5

6

7

▼

```
CREATE INDEX idx_assessment_subject_grade
ON assessment(subject_id, grade);
```

Data Output

Сообщения

Notifications

```
CREATE INDEX
```

Запрос завершён успешно, время выполнения: 110 msec.

### 3. Выполнение тех же запросов с индексами

Запросы были повторно выполнены, планы запросов с помощью EXPLAIN ANALYZE показали изменение стратегии доступа — в том числе использование индексного поиска (Index Scan или Bitmap Index Scan).

Запрос

История запросов

1

2

3

4

5

6

7

EXPLAIN ANALYZE

SELECT \*

FROM assessment

WHERE subject\_id = 101;

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

QUERY PLAN

text

🔒

1

2

3

4

5

Seq Scan on assessment (cost=0.00..1.43 rows=1 width=24) (actual time=0.011..0.014 rows=9 loops=...

Filter: (subject\_id = 101)

Rows Removed by Filter: 25

Planning Time: 0.693 ms

Execution Time: 0.036 ms

Запрос

История запросов

1

2

3

4

5

6

7

EXPLAIN ANALYZE

SELECT \*

FROM assessment

WHERE subject\_id = 101 AND grade = 5;

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

QUERY PLAN

text

🔒

1

2

3

4

5

Seq Scan on assessment (cost=0.00..1.51 rows=1 width=24) (actual time=0.009..0.012 rows=2 loops=...

Filter: ((subject\_id = 101) AND (grade = '5':double precision))

Rows Removed by Filter: 32

Planning Time: 0.633 ms

Execution Time: 0.021 ms

Индекс помог оптимизировать доступ к данным, уменьшив время выборки примерно вдвое.

## Удаление индексов

Запрос	История запросов
1	<b>DROP INDEX IF EXISTS</b> idx_assessment_subject_id;
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Data Output	Сообщения	Notifications
DROP INDEX		
Запрос завершён успешно, время выполнения: 154 msec.		

Запрос

История запросов

1

DROP INDEX IF EXISTS idx\_assessment\_subject\_grade;

2

3

4

5

6

7

8

9

10

11

12

13

Data Output

Сообщения

Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 133 msec.

#### 4. Вывод по лабораторной работе:

В ходе выполнения лабораторной работы были приобретены практические навыки разработки и выполнения сложных SQL-запросов на выборку и модификацию данных в реляционной базе PostgreSQL.