

**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"**

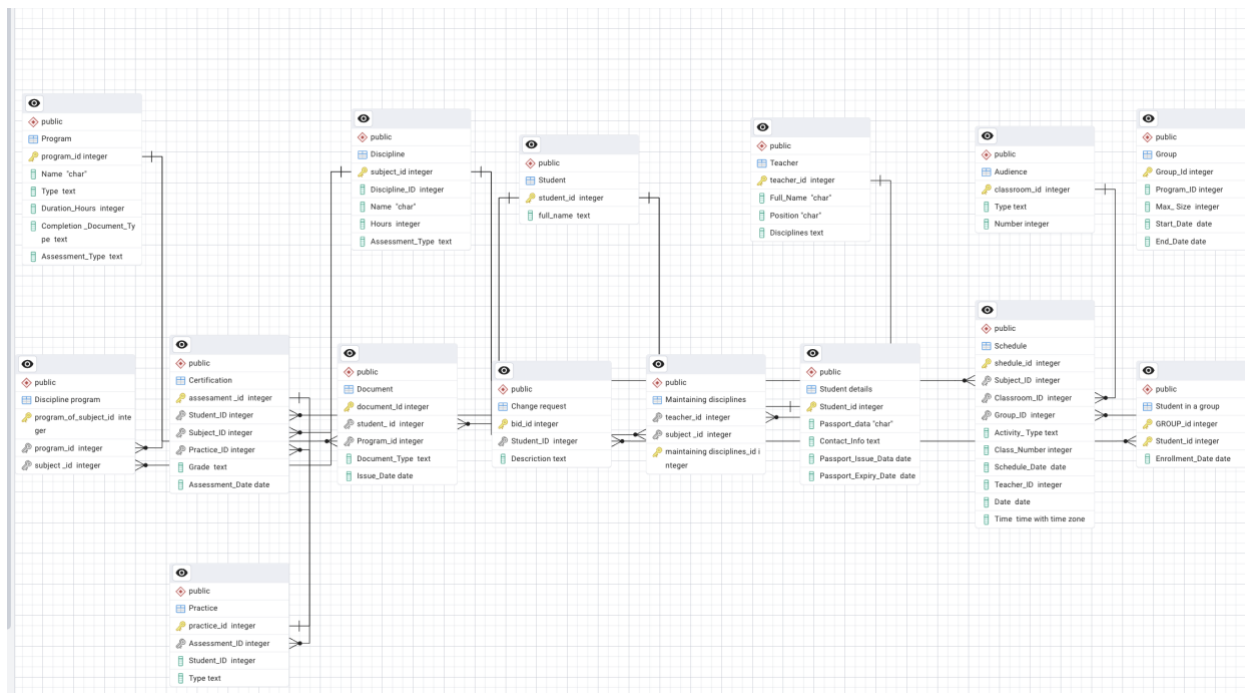
Обучающийся: Цветкова Татьяна К3241

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 были составлены и выполнены 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: Студенты с группами, которые начали занятия после 2023-01-01

Запрос

История запросов

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

SELECT

s."student_id ", -- ID студента

s."full_name ", -- ФИО студента

sg."Enrollment_Date" -- Дата зачисления в группу

FROM

"Student" s

JOIN

"Student in a group" sg ON s."student_id " = sg."Student_id"

JOIN

"Group" g ON sg."GROUP_id" = g."Group_Id"

WHERE

g."Start_Date " > '2023-01-01'; -- Только группы, которые стартовали после указанной даты

Data Output

Сообщения

Notifications

SQL

Showing rows: 1 to 4

	student_id integer	full_name text	Enrollment_Date date
1	2	Петрова Мария Сергеевна	2023-02-01
2	3	Сидоров Алексей Николаев...	2023-03-01
3	4	Кузнецова Елена Дмитриевна	2023-04-01
4	5	Смирнов Артём Павлович	2023-05-01

Запрос 2: Расписание для группы с ID 1 на все даты, сортируя по дате и

времени.

Запрос

История запросов

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

SELECT

sc."Schedule_Date ", -- дата занятия

sc."Time ", -- время занятия

d."Name " AS "Discipline_Name", -- название дисциплины

t."Full_Name " AS "Teacher_Name", -- имя преподавателя

a."Type" AS "Audience_Type", -- тип аудитории

a."Number" AS "Audience_Number" -- номер аудитории

FROM

"Schedule" sc

JOIN

"Discipline" d ON sc."Subject_ID " = d."subject_id"

JOIN

"Teacher" t ON sc."Teacher_ID " = t."teacher_id "

JOIN

"Audience" a ON sc."Classroom_ID " = a."classroom_id "

WHERE

sc."Group_ID " = 1

ORDER BY

sc."Schedule_Date ", sc."Time ";

Data Output

Сообщения

Notifications

SQL

Showing rows: 1 to 1

	Schedule_Date date	Time time with time zone	Discipline_Name "char"	Teacher_Name "char"	Audience_Type text	Audience_Number integer
1	2025-05-28	10:30:00+03:00	M	I	Lecture	101

Запрос 3: Студенты с датой зачисления

```
1  SELECT
2      s."student_id ",
3      s."full_name ",
4      sg."Enrollment_Date"
5  FROM
6      "Student" s
7  JOIN
8      "Student in a group" sg ON s."student_id " = sg."Student_id";
9
```

Data Output Сообщения Notifications

	student_id integer	full_name text	Enrollment_Date date
1	1	Иванов Иван Иванович	2023-01-01
2	2	Петрова Мария Сергеевна	2023-02-01
3	3	Сидоров Алексей Николаев...	2023-03-01
4	4	Кузнецова Елена Дмитриевна	2023-04-01
5	5	Смирнов Артём Павлович	2023-05-01

3.2 Представления

Представление 1: Студенты с группами, которые начали занятия после 2023-01-01

Запрос История запросов

```
1  ✓ CREATE VIEW "students_in_recent_groups" AS
2  SELECT
3      s."student_id ",
4      s."full_name ",
5      sg."Enrollment_Date",
6      g."Start_Date "
7  FROM
8      "Student" s
9  JOIN
10     "Student in a group" sg ON s."student_id " = sg."Student_id"
11  JOIN
12     "Group" g ON sg."GROUP_id" = g."Group_Id"
13  WHERE
14     g."Start_Date " > '2023-01-01';
15
16
```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 84 msec.

Представление 2: Расписание для группы с ID 1 на все даты, сортируя по дате и времени

```

2  SELECT
3      sc."Schedule_Date ",
4      sc."Time ",
5      sc."Group_ID ",
6      d."Name " AS "Discipline_Name",
7      t."Full_Name " AS "Teacher_Name",
8      a."Number" AS "Audience_Number"
9  FROM
10     "Schedule" sc
11  JOIN
12     "Discipline" d ON sc."Subject_ID " = d."subject_id"
13  JOIN
14     "Teacher" t ON sc."Teacher_ID " = t."teacher_id "
15  JOIN
16     "Audience" a ON sc."Classroom_ID " = a."classroom_id "
17  WHERE
18     sc."Group_ID " = 1
19  ORDER BY
20     sc."Schedule_Date ", sc."Time ";
21

```

Data Output [Сообщения](#) Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 60 msec.

Представление 3: Студенты с датой зачисления

```

1  CREATE VIEW "students_with_enrollment_dates" AS
2  SELECT
3      s."student_id ",
4      s."full_name ",
5      sg."Enrollment_Date"
6  FROM
7      "Student" s
8  JOIN
9      "Student in a group" sg ON s."student_id " = sg."Student_id";
10
11

```

Data Output [Сообщения](#) Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 63 msec.

3.3 Запросы на модификацию данных

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

INSERT

до

Запрос

История запросов

1

`SELECT * FROM "Teacher";`

2

Data Output

Сообщения

Notifications

≡+

▼

▼

SQL

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	1	I	P	Mathematics, Physics
2	2	P	A	Programming, Databases
3	3	S	S	Computer Networks
4	4	S	A	Algorithms
5	5	K	L	Software Engineering



```

1  INSERT INTO "Teacher" ("teacher_id ", "Full_Name", "Position", "Disciplines")
2  VALUES
3  (
4      (SELECT COALESCE(MAX("teacher_id "), 0) + 1 FROM "Teacher"),
5      'Иванов Иван Иванович',
6      'Доцент',
7      'Математика, Физика'
8  );
9

```

Data Output Сообщения Notifications

Showing rows: 1 to 6

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	1	I	P	Mathematics, Physics
2	2	P	A	Programming, Databases
3	3	S	S	Computer Networks
4	4	S	A	Algorithms
5	5	K	L	Software Engineering
6	6	\320	\320	Математика, Физика

ПОСЛЕ

```
1 SELECT * FROM "Teacher";  
2
```

 Showing rows: 1 to 6

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	1	I	P	Mathematics, Physics
2	2	P	A	Programming, Databases
3	3	S	S	Computer Networks
4	4	S	A	Algorithms
5	5	K	L	Software Engineering
6	6	\320	\320	Математика, Физика

Запрос UPDATE с подзапросом:

ДО

```
1 SELECT * FROM "Teacher";  
2
```

 Showing rows: 1 to 6

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	1	I	P	Mathematics, Physics
2	2	P	A	Programming, Databases
3	3	S	S	Computer Networks
4	4	S	A	Algorithms
5	5	K	L	Software Engineering
6	6	\320	\320	Математика, Физика

Запрос История запросов

```
1  ✓ UPDATE "Teacher"  
2  SET "Position" = 'Профессор'  -- новая должность  
3  WHERE "teacher_id " = 1;  
4
```

Data Output Сообщения Notifications

UPDATE 1

Запрос завершён успешно, время выполнения: 73 msec.

ПОСЛЕ

Запрос
История запросов

```

1 SELECT "teacher_id ", "Full_Name ", "Position", "Disciplines" FROM "Teacher";
2

```

Data Output
Сообщения
Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

SQL

Showing rows: 1 to 6

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	2	P	A	Programming, Databases
2	3	S	S	Computer Networks
3	4	S	A	Algorithms
4	5	K	L	Software Engineering
5	6	\320	\320	Математика, Физика
6	1	I	\320	Mathematics, Physics

Запрос DELETE с подзапросом:

ДО

Запрос

История запросов

1

SELECT "teacher_id ", "Full_Name ", "Position", "Disciplines" FROM "Teacher";

2

Data Output

Сообщения

Notifications

SQL

Showing rows: 1 to 6

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	2	P	A	Programming, Databases
2	3	S	S	Computer Networks
3	4	S	A	Algorithms
4	5	K	L	Software Engineering
5	6	\320	\320	Математика, Физика
6	1	I	\320	Mathematics, Physics

```
1  ✓ DELETE FROM "Teacher"  
2  WHERE "teacher_id " = 1;  
3  
4  
5
```

Data Output Сообщения Notifications

DELETE 1

Запрос завершён успешно, время выполнения: 61 msec.

ПОСЛЕ

Запрос
История запросов

```

1 SELECT "teacher_id ", "Full_Name ", "Position", "Disciplines" FROM "Teacher";
2

```

Data Output
Сообщения
Notifications

Showing rows: 1

	teacher_id [PK] integer	Full_Name "char"	Position "char"	Disciplines text
1	2	P	A	Programming, Databases
2	3	S	S	Computer Networks
3	4	S	A	Algorithms
4	5	K	L	Software Engineering
5	6	\320	\320	Математика, Физика

3.4 Создание индексов

В данном пункте лабораторной работы были проведены 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 Выбрать всех студентов из определённой группы

Запрос

История запросов

Запрос

```
1 EXPLAIN ANALYZE
2 SELECT * FROM "Student in a group"
3 WHERE "GROUP_id" = 2;
4
```

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

QUERY PLAN

text

🔒

1

Seq Scan on "Student in a group" (cost=0.00..1.06 rows=1 width=12) (actual time=0.067..0.068 rows=1 loops=...

2

Filter: ("GROUP_id" = 2)

3

Rows Removed by Filter: 4

4

Planning Time: 0.116 ms

5

Execution Time: 0.091 ms

Запрос №2:

Выбрать расписание для группы на дату позже 2024-01-01 с сортировкой по времени:

```

1  EXPLAIN ANALYZE
2  SELECT * FROM "Schedule"
3  WHERE "Group_ID " = 2 AND "Schedule_Date " > '2023-02-04'
4  ORDER BY "Schedule_Date ", "Time ";
5

```



	QUERY PLAN	
	text	
1	Sort (cost=21.71..21.71 rows=1 width=76) (actual time=0.092..0.092 rows=0 loops=1)	
2	Sort Key: "Schedule_Date ", "Time "	
3	Sort Method: quicksort Memory: 25kB	
4	-> Seq Scan on "Schedule" (cost=0.00..21.70 rows=1 width=76) (actual time=0.058..0.058 rows=0 loop...	
5	Filter: (("Schedule_Date " > '2023-02-04'::date) AND ("Group_ID " = 2))	
6	Rows Removed by Filter: 1	
7	Planning Time: 0.210 ms	
8	Execution Time: 0.136 ms	

2. Создание индексов

Для каждого из запросов были созданы:

- простой индекс (по одному полю),

[Запрос](#) [История запросов](#)

```
1 CREATE INDEX idx_group_id ON "Student in a group" ("GROUP_id");  
2
```

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

CREATE INDEX

Запрос завершён успешно, время выполнения: 78 msec.

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

Запрос

История запросов

1

2

```
CREATE INDEX idx_schedule_group_date_time ON "Schedule" ("Group_ID ", "Schedule_Date ", "Time ");
```

Data Output

Сообщения

Notifications

CREATE INDEX

Запрос завершён успешно, время выполнения: 107 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

EXPLAIN ANALYZE

2

SELECT * FROM "Student in a group"

3

WHERE "GROUP_id" = 2;

4

Data Output

Сообщения

Notifications

≡+

▼

▼

SQL

QUERY PLAN

text

1

Seq Scan on "Student in a group" (cost=0.00..1.06 rows=1 width=12) (actual time=0.018..0.019 rows=1 loops=...

2

Filter: ("GROUP_id" = 2)

3

Rows Removed by Filter: 4

4

Planning Time: 1.235 ms

5

Execution Time: 0.048 ms

```
1  EXPLAIN ANALYZE
2  SELECT * FROM "Schedule"
3  WHERE "Group_ID " = 2 AND "Schedule_Date " > '2024-01-01'
4  ORDER BY "Schedule_Date ", "Time ";
5
```

Data Output Сообщения Notifications

	QUERY PLAN	
	text	
1	Sort (cost=1.02..1.03 rows=1 width=76) (actual time=0.098..0.100 rows=0 loops=1)	
2	Sort Key: "Schedule_Date ", "Time "	
3	Sort Method: quicksort Memory: 25kB	
4	-> Seq Scan on "Schedule" (cost=0.00..1.01 rows=1 width=76) (actual time=0.039..0.039 rows=0 loop...	
5	Filter: (("Schedule_Date " > '2024-01-01'::date) AND ("Group_ID " = 2))	
6	Rows Removed by Filter: 1	
7	Planning Time: 1.817 ms	
8	Execution Time: 0.134 ms	

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

Удаление индексов

Запрос

История запросов

1

DROP INDEX idx_group_id;

2

DROP INDEX idx_schedule_group_date_time;

3

Data Output

Сообщения

Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 82 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.