

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»**

**Отчет**

по лабораторной работе №2

«Запросы на выборку и модификацию данных, представления и  
индексы в PostgreSQL»  
по дисциплине «Проектирование и реализация баз данных»

Вариант №7. БД «Курсы»

Авторы: Антонова М. А.

Факультет: ФИКТ

Группа: К32422

Преподаватель: Говорова М.М.

Санкт-Петербург, 2023

## **1 Описание работы**

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

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

**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) 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.

## **2 Описание выполнения работы**

- 1) запросы к базе данных. Выполнить запросы согласно индивидуальному заданию, часть 2. Необходимо привести формулировку запроса, команду, скриншот выполнения запроса;
- 2) представления. Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3. Необходимо привести

формулировку запроса, команду, скриншот выполнения запроса и просмотр содержимого представления;

- 3) запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). Необходимо привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность 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.

### **3 Описание предметной области**

#### **Вариант 7. БД «Курсы»**

Сеть учебных подразделений занимается организацией внебюджетного образования.

Имеется несколько образовательных программ краткосрочных курсов, предназначенных для определенных специальностей, связанных с программным обеспечением ИТ. Каждая программа имеет определенную длительность и свой перечень изучаемых дисциплин. Одна дисциплина может относиться к нескольким программам. На каждую программу может быть набрано несколько групп обучающихся. По каждой дисциплине могут проводиться лекционные, лабораторные/практические занятия и практика определенном объеме часов. По каждой дисциплине и практике проводится аттестация в формате экзамен/дифзачет/зачет.

Подразделение обеспечивает следующие ресурсы: учебные классы, лекционные аудитории и преподавателей. Необходимо составить расписание занятий.

БД должна содержать следующий минимальный набор сведений: Фамилия слушателя. Имя слушателя. Паспортные данные. Контакты. Код программы. Программа. Тип программы. Объем часов. Номер группы. максимальное количество человек в группе (для набора). Дата начала обучения. Дата окончания обучения. Название дисциплины. Количество часов. Дата занятий. Номер пары. Номер аудитории. Тип аудитории. Адрес площадки. Вид занятий (лекционные, практические или лабораторные). Фамилия преподавателя. Имя и отчество преподавателя. Должность преподавателя. Дисциплины, которые может вести преподаватель.



FROM "Admission\_year"

LEFT JOIN "Programme" ON "Programme"."ID\_programme"="Admission\_year"."a\_ID\_programme"

LEFT JOIN "Group" ON "Admission\_year"."ID\_year\_admission"="Group"."g\_ID\_year\_admission"

WHERE "Group"."Number\_of\_students" < 10

The screenshot shows a database query editor with a 'Query' tab selected. The query is as follows:

```
1 SELECT "Group"."Group_number", "Programme"."Name_of_programme"
2 FROM "Admission_year"
3 LEFT JOIN "Programme" ON "Programme"."ID_programme"="Admission_year"."a_ID_programme"
4 LEFT JOIN "Group" ON "Admission_year"."ID_year_admission"="Group"."g_ID_year_admission"
5 WHERE "Group"."Number_of_students" < 10
6
7
```

Below the query editor, there is a 'Data Output' tab showing the results of the query. The results are displayed in a table with two columns: 'Group\_number' (integer) and 'Name\_of\_programme' (character varying (50)).

	Group_number integer	Name_of_programme character varying (50)
1	1112020	Филология
2	1262018	Астрономия

2) Вывести список преподавателей с указанием количества программ, где они преподавали за истекший год.

SELECT "Teacher"."Full\_name", COUNT("Discipline\_of\_programme"."d\_ID\_programme")

FROM "Lesson"

LEFT JOIN "Discipline\_of\_programme" ON "Lesson"."L\_ID\_discipline\_of\_pr"  
="Discipline\_of\_programme"."ID\_discipline\_of\_pr"

LEFT JOIN "Teacher" ON "Lesson"."L\_ID\_passport"="Teacher"."ID\_passport"

WHERE "Lesson"."Date\_of\_lessons" > (NOW() - INTERVAL '1 YEAR') AND  
"Lesson"."Date\_of\_lessons"<NOW()

GROUP BY "Teacher"."Full\_name"

Query
Query History

```

1 SELECT "Teacher"."Full_name", COUNT("Discipline_of_programme"."d_ID_programme")
2 FROM "Lesson"
3 LEFT JOIN "Discipline_of_programme" ON "Lesson"."L_ID_discipline_of_pr" ="Discipline_o
4 LEFT JOIN "Teacher" ON "Lesson"."L_ID_passport"="Teacher"."ID_passport"
5 WHERE "Lesson"."Date_of_lessons" > (NOW() - INTERVAL '1 YEAR') AND "Lesson"."Date_of_l
6 GROUP BY "Teacher"."Full_name"
7
8

```

Data Output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	Full_name character varying (50)	count bigint
1	Петров Николай Владимирович	2
2	Смирнова Елена Васильевна	1

3) Вывести список преподавателей, которые не проводят занятия на третьей паре ни в один из дней недели.

```

SELECT DISTINCT "Lesson"."L_ID_passport" FROM "Lesson"
WHERE "Lesson"."L_ID_passport" NOT IN (SELECT
"Lesson"."L_ID_passport" FROM
"Lesson" WHERE "Lesson"."Lesson_number" = '3')

```

Query Query History

```

1
2 SELECT DISTINCT "Lesson"."L_ID_passport" FROM "Lesson"
3 WHERE "Lesson"."L_ID_passport" NOT IN (SELECT
4 "Lesson"."L_ID_passport" FROM
5 "Lesson" WHERE "Lesson"."Lesson_number" = '3')

```

Data Output Messages Notifications

L\_ID\_passport  
character varying (10) 🔒

1	4018785670
---	------------

4) Вывести список свободных лекционных аудиторий на ближайший понедельник.

```

SELECT "Class"."ID_class" FROM "Class"
WHERE "Class"."ID_class" not IN
(SELECT "Lesson"."L_ID_class" FROM "Lesson"
WHERE "Lesson"."Date_of_lessons" = DATE(DATE_TRUNC('week', NOW()) + INTERVAL '7 days'))
AND "Class"."Class_type" = 'лекционная'

```

Query Query History

```

1 SELECT "Class"."ID_class" FROM "Class"
2 WHERE "Class"."ID_class" not IN
3 (SELECT "Lesson"."L_ID_class" FROM "Lesson"
4 WHERE "Lesson"."Date_of_lessons" = DATE(DATE_TRUNC('week', NOW()) + INTERVAL '7 days')
5 AND "Class"."Class_type" = 'лекционная'
6
7

```

Data Output Messages Notifications

ID\_class  
[PK] integer

1	1003
---	------



5) Вычислить общее количество обучающихся по каждой программе за последний год.

```
SELECT p."Name_of_programme",COUNT(s."I_ID_passport")
FROM
    (SELECT * FROM "Is_in_group" as i
    RIGHT JOIN "Group"
    ON "Group"."Group_number" = i."I_group_number"
    RIGHT JOIN "Admission_year"
    ON "Admission_year"."ID_year_admission" = "Group"."g_ID_year_admission"
    where not("Admission_year"."To" < (NOW() - INTERVAL '1 YEAR')
            OR "Admission_year"."From">NOW())) AS s
RIGHT JOIN "Programme" as p ON p."ID_programme" = s."a_ID_programme"
GROUP BY p."Name_of_programme"
```

Query Query History

1 SELECT p."Name\_of\_programme",COUNT(s."I\_ID\_passport")

2 FROM

3 (SELECT \* FROM "Is\_in\_group" as i

4 RIGHT JOIN "Group"

5 ON "Group"."Group\_number" = i."I\_group\_number"

6 RIGHT JOIN "Admission\_year"

7 ON "Admission\_year"."ID\_year\_admission" = "Group"."g\_ID\_year\_admission"

8 where not("Admission\_year"."To" < (NOW() - INTERVAL '1 YEAR')

9 OR "Admission\_year"."From">NOW())) AS s








10



11 RIGHT JOIN "Programme" as p ON p."ID\_programme" = s."a\_ID\_programme"

12 GROUP BY p."Name\_of\_programme"

13

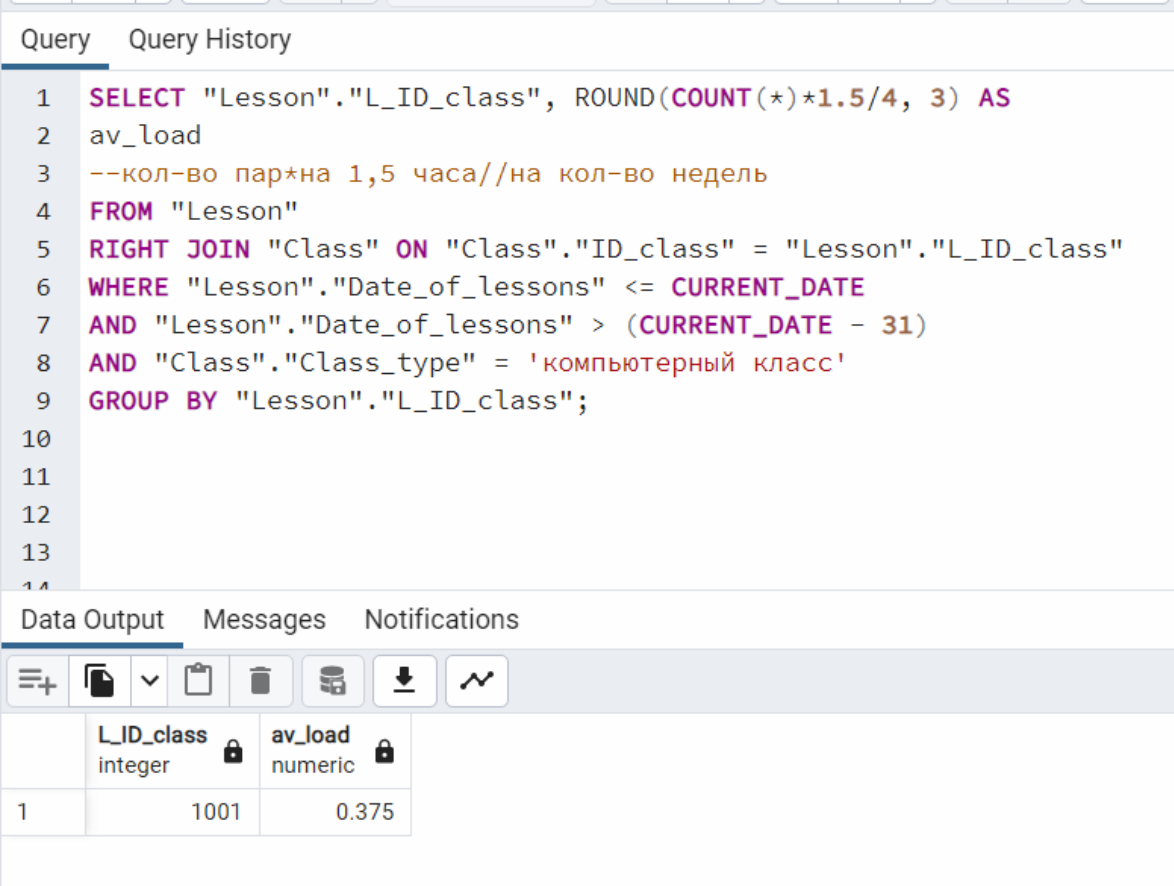
Data Output Messages Notifications



	Name_of_programme character varying (50) 	count bigint 
1	Инноватика	0
2	Филология	3
3	Бизнес-информатика	1
4	Аналитик данных	0
5	Астрономия	1

6) Вычислить среднюю загруженность компьютерных классов в неделю за последний месяц (в часах).

```
SELECT "Lesson"."L_ID_class", ROUND(COUNT(*)*1.5/4, 3) AS  
av_load  
--кол-во пар*на 1,5 часа//на кол-во недель  
FROM "Lesson"  
RIGHT JOIN "Class" ON "Class"."ID_class" = "Lesson"."L_ID_class"  
WHERE "Lesson"."Date_of_lessons" <= CURRENT_DATE  
AND "Lesson"."Date_of_lessons" > (CURRENT_DATE - 31)  
AND "Class"."Class_type" = 'компьютерный класс'  
GROUP BY "Lesson"."L_ID_class";
```



The screenshot shows a database query editor with a tab labeled "Query". The query is as follows:

```
1 SELECT "Lesson"."L_ID_class", ROUND(COUNT(*)*1.5/4, 3) AS  
2 av_load  
3 --кол-во пар*на 1,5 часа//на кол-во недель  
4 FROM "Lesson"  
5 RIGHT JOIN "Class" ON "Class"."ID_class" = "Lesson"."L_ID_class"  
6 WHERE "Lesson"."Date_of_lessons" <= CURRENT_DATE  
7 AND "Lesson"."Date_of_lessons" > (CURRENT_DATE - 31)  
8 AND "Class"."Class_type" = 'компьютерный класс'  
9 GROUP BY "Lesson"."L_ID_class";  
10  
11  
12  
13  
14
```

Below the query editor, there is a "Data Output" tab showing the results of the query. The results are displayed in a table with two columns: "L\_ID\_class" (integer) and "av\_load" (numeric). The table contains one row of data:

	L_ID_class integer	av_load numeric
1	1001	0.375

7) Найти самые популярные программы за последние 3 года.

```
SELECT  
"Programme"."Name_of_programme", SUM("Admission_year"."Number_of_admission_year"),  
"Programme"."Cost", "Programme"."ID_programme"  
FROM "Programme"  
LEFT JOIN "Admission_year" ON "Admission_year"."a_ID_programme" =  
"Programme"."ID_programme"
```

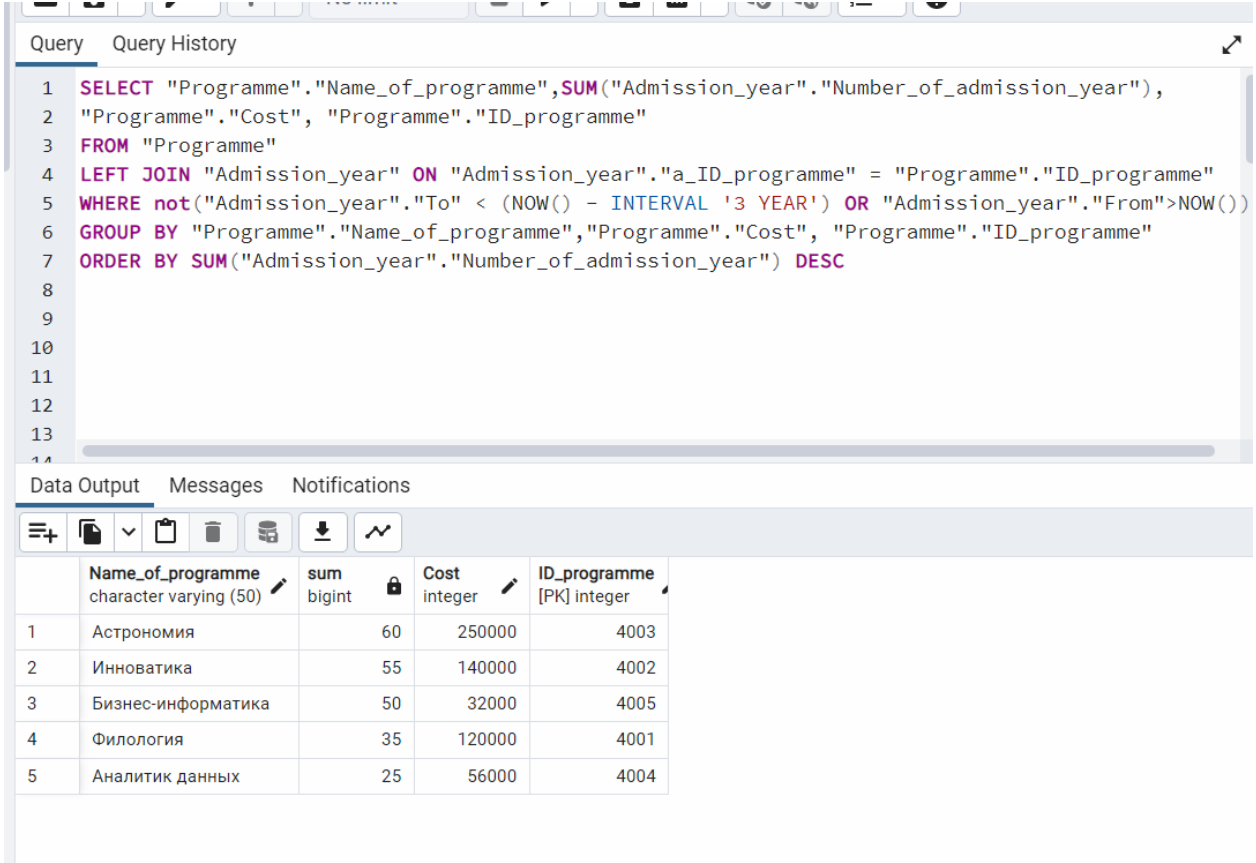
```

WHERE not("Admission_year"."To" < (NOW() - INTERVAL '3 YEAR') OR
"Admission_year"."From">NOW())

GROUP BY "Programme"."Name_of_programme", "Programme"."Cost",
"Programme"."ID_programme"

ORDER BY SUM("Admission_year"."Number_of_admission_year") DESC

```



The screenshot shows a database query editor with a SQL query in the 'Query' tab. The query is a SELECT statement that joins the 'Programme' table with the 'Admission\_year' table. It filters for admissions within the last 3 years, groups by programme name, cost, and ID, and orders the results by the sum of admission years in descending order. Below the query editor, the 'Data Output' tab shows the results of the query as a table with 5 rows and 5 columns.

	Name_of_programme character varying (50)	sum bigint	Cost integer	ID_programme [PK] integer
1	Астрономия	60	250000	4003
2	Иноватика	55	140000	4002
3	Бизнес-информатика	50	32000	4005
4	Филология	35	120000	4001
5	Аналитик данных	25	56000	4004

### Задание 3. Создать представления:

1. для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов;

```

CREATE VIEW "All_Programmes2" AS

SELECT "Programme"."Name_of_programme",

"Discipline"."Name_of_discipline" AS "Disciplines",

"Discipline"."Sum_hours" as "Total_hours"

FROM "Discipline_of_programme"

LEFT JOIN "Discipline" ON "Discipline"."ID_discipline" =

"Discipline_of_programme"."d_ID_discipline"

```

```

LEFT JOIN "Programme" ON "Programme"."ID_programme" =
"Discipline_of_programme"."d_ID_programme"

GROUP BY "Programme"."Name_of_programme",

"Discipline"."Name_of_discipline", "Total_hours"

ORDER BY "Programme"."Name_of_programme",

"Discipline"."Name_of_discipline" DESC

```

The screenshot shows a database query editor with a SQL query and its results. The query creates a view named 'All\_Programmes2' by joining 'Programme' and 'Discipline' tables. The results table shows the data for the first five programs.

**Query:**

```

1 CREATE VIEW "All_Programmes2" AS
2 SELECT "Programme"."Name_of_programme",
3 "Discipline"."Name_of_discipline" AS "Disciplines",
4 "Discipline"."Sum_hours" as "Total_hours"
5 FROM "Discipline_of_programme"
6 LEFT JOIN "Discipline" ON "Discipline"."ID_discipline" = "Discipline_of_programme"."d_ID_discipl
7 LEFT JOIN "Programme" ON "Programme"."ID_programme" = "Discipline_of_programme"."d_ID_programme"
8 GROUP BY "Programme"."Name_of_programme",
9 "Discipline"."Name_of_discipline", "Total_hours"
10 ORDER BY "Programme"."Name_of_programme",
11 "Discipline"."Name_of_discipline" DESC

```

**Data Output:**

	Name_of_programme character varying (50)	sum bigint	Cost integer	ID_programme [PK] integer
1	Астрономия	60	250000	4003
2	Инноватика	55	140000	4002
3	Бизнес-информатика	50	32000	4005
4	Филология	35	120000	4001
5	Аналитик данных	25	56000	4004

## 2. общий доход по каждой программе за последний год.

```

CREATE VIEW "Income" AS

SELECT "Programme"."Name_of_programme", "Programme"."Cost" AS

cost

FROM "Programme"

GROUP BY "Programme"."Name_of_programme", "Programme"."Cost"

ORDER BY cost DESC

```



Query Query History

```
1 CREATE VIEW "Income" AS
2 SELECT "Programme"."Name_of_programme", "Programme"."Cost" AS
3 cost
4 FROM "Programme"
5 GROUP BY "Programme"."Name_of_programme", "Programme"."Cost"
6 ORDER BY cost DESC
7
8
```

Query Query History

```
1 SELECT * FROM timetable_courses."Income"
2
```

Data Output Messages Notifications

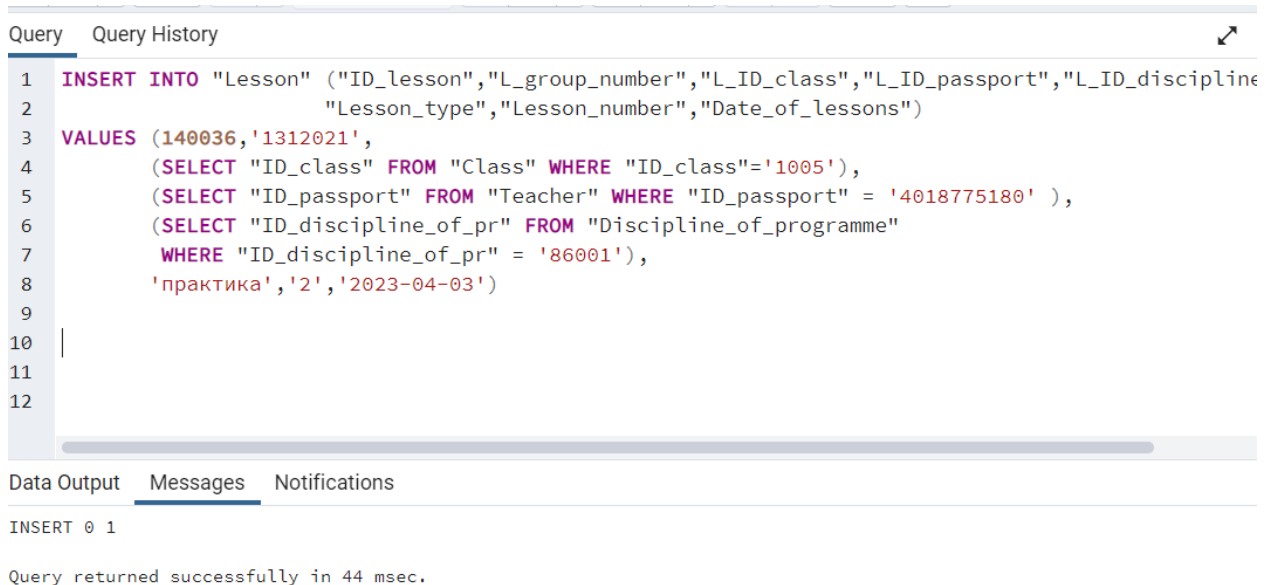
	Name_of_programme character varying (50) 	cost integer 
1	Астрономия	250000
2	Инноватика	140000
3	Филология	120000
4	Аналитик данных	56000
5	Бизнес-информатика	32000

## Задание 4 – запросы на модификацию данных

### Запрос “INSERT”

1. Добавить группе 1312021 в расписание практическое занятие по дисциплине математика 3 апреля 2023 года

```
INSERT INTO "Lesson"
("ID_lesson","L_group_number","L_ID_class","L_ID_passport","L_ID_discipline_of_pr",
"Lesson_type","Lesson_number","Date_of_lessons")
VALUES (140036,'1312021',
(SELECT "ID_class" FROM "Class" WHERE "ID_class"='1005'),
(SELECT "ID_passport" FROM "Teacher" WHERE "ID_passport" = '4018775180' ),
(SELECT "ID_discipline_of_pr" FROM "Discipline_of_programme"
WHERE "ID_discipline_of_pr" = '86001'),
'практика','2','2023-04-03')
```



The screenshot shows a database query editor with a tab labeled "Query". The query text is as follows:

```
1 INSERT INTO "Lesson" ("ID_lesson","L_group_number","L_ID_class","L_ID_passport","L_ID_discipline_of_pr",
2 "Lesson_type","Lesson_number","Date_of_lessons")
3 VALUES (140036,'1312021',
4 (SELECT "ID_class" FROM "Class" WHERE "ID_class"='1005'),
5 (SELECT "ID_passport" FROM "Teacher" WHERE "ID_passport" = '4018775180' ),
6 (SELECT "ID_discipline_of_pr" FROM "Discipline_of_programme"
7 WHERE "ID_discipline_of_pr" = '86001'),
8 'практика','2','2023-04-03')
9
10
11
12
```

Below the query editor, there are three tabs: "Data Output", "Messages", and "Notifications". The "Messages" tab is selected, showing the following output:

```
INSERT 0 1
```

At the bottom, a status message reads: "Query returned successfully in 44 msec."

До:

1

SELECT \* FROM timetable\_courses."Lesson"

2

ORDER BY "ID\_lesson" ASC

	ID_lesson [PK] integer	L_group_number integer	L_ID_class integer	L_ID_passport character varying (10)	L_ID_discipline_of_pr integer	Lesson_type timetable_courses."Lesson_type"	Lesson_number integer	Date_of_lesson date
1	140031	1112020	1002	4018785670	86003	практика	1	2023-02-02
2	140032	1262018	1002	4018775180	86002	лекция	2	2023-03-02
3	140033	1312021	1005	4018775180	86001	лабораторная	3	2023-01-02
4	140034	1112020	1005	4018785670	86001	лекция	2	2023-03-27
5	140035	1262018	1001	4018785670	86003	практика	4	2023-03-25

После:

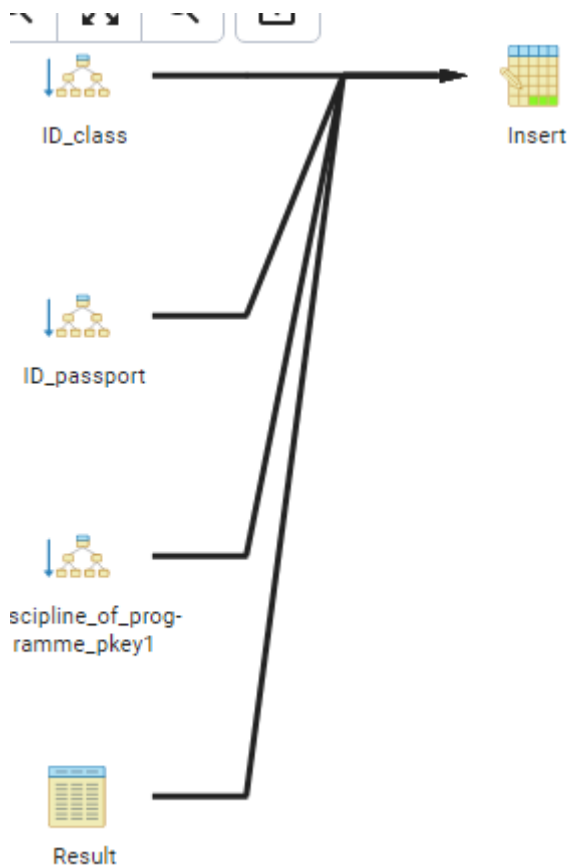
1

SELECT \* FROM timetable\_courses."Lesson"

2

ORDER BY "ID\_lesson" ASC

	ID_lesson [PK] integer	L_group_number integer	L_ID_class integer	L_ID_passport character varying (10)	L_ID_discipline_of_pr integer	Lesson_type timetable_courses."Lesson_type"	Lesson_number integer	Date_of_lesson date
1	140031	1112020	1002	4018785670	86003	практика	1	2023-02-02
2	140032	1262018	1002	4018775180	86002	лекция	2	2023-03-02
3	140033	1312021	1005	4018775180	86001	лабораторная	3	2023-01-02
4	140034	1112020	1005	4018785670	86001	лекция	2	2023-03-27
5	140035	1262018	1001	4018785670	86003	практика	4	2023-03-25
6	140036	1312021	1005	4018775180	86001	практика	2	2023-04-03



## 2. Отменить второе занятие 27 марта у группы 1112020

DELETE FROM "Lesson" WHERE "Lesson"."ID\_lesson" IN

(SELECT "Lesson"."ID\_lesson" FROM "Lesson" WHERE "L\_group\_number"='1112020'

AND "Date\_of\_lessons" = '2023-03-27')

Query	Query History
5	DELETE FROM "Lesson" WHERE "Lesson"."ID_lesson" IN
6	(SELECT "Lesson"."ID_lesson" FROM "Lesson" WHERE "L_group_number"='1112020'
7	AND "Date_of_lessons" = '2023-03-27')
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	

Data Output	Messages	Notifications
DELETE 1		
Query returned successfully in 42 msec.		



До:

QueryQuery History

1SELECT \* FROM timetable\_courses."Lesson"

2ORDER BY "ID\_lesson" ASC

Scratch Pad

	ID_lesson [PK] integer	L_group_number integer	L_ID_class integer	L_ID_passport character varying (10)	L_ID_discipline_of_pr integer	Lesson_type timetable_courses."Lesson_type"	Lesson_number integer	Date_of_lesson date
1	140031	1112020	1002	4018785670	86003	практика	1	2023-02-02
2	140032	1262018	1002	4018775180	86002	лекция	2	2023-03-02
3	140033	1312021	1005	4018775180	86001	лабораторная	3	2023-01-02
4	140034	1112020	1005	4018785670	86001	лекция	2	2023-03-27
5	140035	1262018	1001	4018785670	86003	практика	4	2023-03-25
6	140036	1312021	1005	4018775180	86001	практика	2	2023-04-03

После:

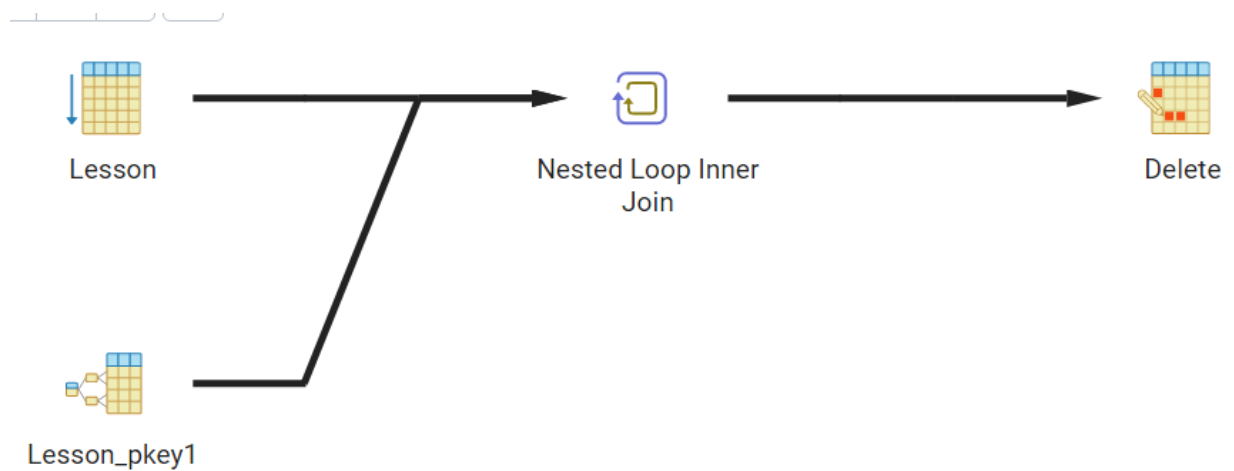
QueryQuery History

1SELECT \* FROM timetable\_courses."Lesson"

2ORDER BY "ID\_lesson" ASC

Scratch Pad

	ID_lesson [PK] integer	L_group_number integer	L_ID_class integer	L_ID_passport character varying (10)	L_ID_discipline_of_pr integer	Lesson_type timetable_courses."Lesson_type"	Lesson_number integer	Date_of_lesson date
1	140031	1112020	1002	4018785670	86003	практика	1	2023-02-02
2	140032	1262018	1002	4018775180	86002	лекция	2	2023-03-02
3	140033	1312021	1005	4018775180	86001	лабораторная	3	2023-01-02
4	140035	1262018	1001	4018785670	86003	практика	4	2023-03-25
5	140036	1312021	1005	4018775180	86001	практика	2	2023-04-03



3. Запрос UPDATE Продлить больничный лаборанту до 8 апреля 2023г  
(Анисимову Денису Петровичу)

```
UPDATE "Sick_and_vacation_leaves" SET "To"='2023-04-08' WHERE "s_ID_passport" IN
(SELECT "Teacher"."ID_passport" FROM "Teacher" WHERE "Position"='лаборант')
```

Query Query History

```
1 UPDATE "Sick_and_vacation_leaves" SET "To"='2023-04-08' WHERE "s_ID_passport" IN
2 (SELECT "Teacher"."ID_passport" FROM "Teacher" WHERE "Position"='лаборант')
3
4
5
6
7
8
9
10
11
12
13
14
```

Data Output Messages Notifications

UPDATE 2

Query returned successfully in 35 msec.

До:

```
1 SELECT * FROM timetable_courses."Sick_and_vacation_leaves"
2 ORDER BY "ID_absense" ASC
```

Data Output Messages Notifications

	ID_absense [PK] integer	From date	To date	Cause_of_absence timetable_courses."Cause_of_absence"	s_ID_passport character varying (10)
1	68861	2023-01-02	2023-01-16	больничный	3018485570
2	68862	2023-02-01	2023-02-15	больничный	4012783672
3	68863	2023-02-20	2023-03-01	отпуск	4017783319
4	68864	2023-03-20	2023-04-01	больничный	4012783672

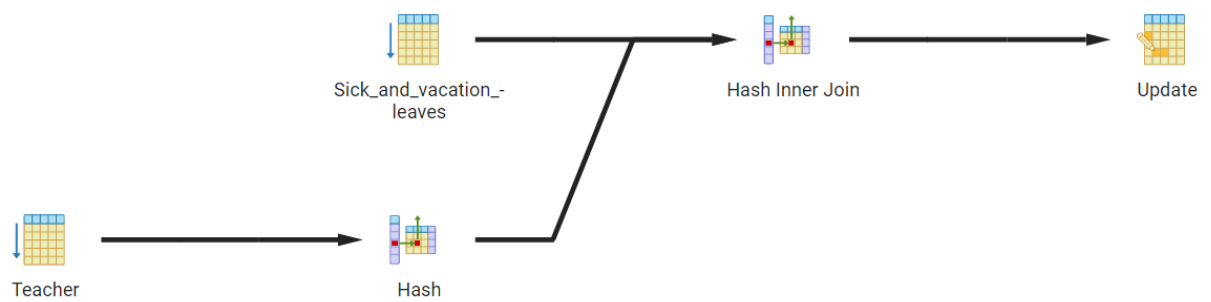
После:

Query Query History

```
1 SELECT * FROM timetable_courses."Sick_and_vacation_leaves"
2 ORDER BY "ID_absense" ASC
```

Data Output Messages Notifications

	ID_absense [PK] integer	From date	To date	Cause_of_absence timetable_courses."Cause_of_absence"	s_ID_passport character varying (10)
1	68861	2023-01-02	2023-01-16	больничный	3018485570
2	68862	2023-02-01	2023-04-08	больничный	4012783672
3	68863	2023-02-20	2023-03-01	отпуск	4017783319
4	68864	2023-03-20	2023-04-08	больничный	4012783672



## Создание индексов

### 1) Запрос без индекса

```
2  
3 SELECT "Class"."ID_class" FROM "Class"  
4 WHERE "Class"."ID_class" not IN  
5 (SELECT "Lesson"."L_ID_class" FROM "Lesson"  
6 WHERE "Lesson"."Date_of_lessons" = DATE(DATE_TRUNC('week', NOW()) + INTERVAL '7 days'))  
7 AND "Class"."Class_type" = 'лекционная'  
8
```

Data Output Messages Notifications

Successfully run. Total query runtime: 346 msec.

### Запрос с индексом

```
CREATE INDEX sv_classes ON "Class" ("ID_class");
```

Query    Query History

```
1 CREATE INDEX sv_classes ON "Class" ("ID_class");
2
3
4
5
6
7
8
9
10
11
12
13
```

Data Output    Messages    Notifications

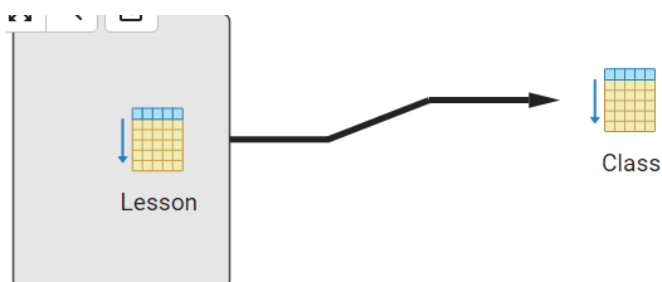
CREATE INDEX

Query returned successfully in 54 msec.

```
1 SELECT "Class"."ID_class" FROM "Class"
2 WHERE "Class"."ID_class" not IN
3 (SELECT "Lesson"."L_ID_class" FROM "Lesson"
4 WHERE "Lesson"."Date_of_lessons" = DATE(DATE_TRUNC('week', NOW()) + INTERVAL '7 days'))
5 AND "Class"."Class_type" = 'лекционная'
6
7
8
9
```

Data Output    Messages    Notifications

Successfully run. Total query runtime: 36 msec.  
1 rows affected.



#	Node
1.	→ Seq Scan on Class as Class Filter: ((NOT (hashed SubPlan 1)) AND ("Class_type" = 'лекционная'::"Class_type"))
2.	→ Seq Scan on Lesson as Lesson Filter: ("Date_of_lessons" = date((date_trunc('week'::text, now()) + '7 days'::interval)))

Общее время выполнения запроса без индекса: 346 msec

Общее время выполнения запроса с индексом: 54 msec

## 2) Запрос без индекса

Query

Query History

1

2

3

4

5

6

7

8

9

SELECT "Group"."Group\_number", "Programme"."Name\_of\_programme"

FROM "Admission\_year"

LEFT JOIN "Programme" ON "Programme"."ID\_programme"="Admission\_year"."a\_ID\_programme"

LEFT JOIN "Group" ON "Admission\_year"."ID\_year\_admission"="Group"."g\_ID\_year\_admission"

WHERE "Group"."Number\_of\_students" < 10

Data Output

Messages

Notifications

Successfully run. Total query runtime: 36 msec.  
2 rows affected.

## Запрос с индексом

CREATE INDEX small\_num\_students ON "Group" ("Group\_number");

Query

Query History

1

2

3

4

5

6

7

8

9

CREATE INDEX small\_num\_students ON "Group" ("Group\_number");

Data Output

Messages

Notifications

CREATE INDEX  
  
Query returned successfully in 31 msec.

Query Query History

```
11 SELECT "Group"."Group_number", "Programme"."Name_of_programme"
12 FROM "Admission_year"
13 LEFT JOIN "Programme" ON "Programme"."ID_programme"="Admission_year"."a_ID_programme"
14 LEFT JOIN "Group" ON "Admission_year"."ID_year_admission"="Group"."g_ID_year_admission"
15 WHERE "Group"."Number_of_students" < 10
16
17
18
19
```

Data Output Messages Notifications

Successfully run. Total query runtime: 33 msec.  
2 rows affected.

Общее время выполнения запроса без индекса: 36 msec

Общее время выполнения запроса с индексом: 33 msec

Запрос без индекса

```
1
2 SELECT "Lesson"."L_ID_class", ROUND(COUNT(*)*1.5/4, 3) AS
3 av_load
4 --кол-во пар*на 1,5 часа//на кол-во недель
5 FROM "Lesson"
6 RIGHT JOIN "Class" ON "Class"."ID_class" = "Lesson"."L_ID_class"
7 WHERE "Lesson"."Date_of_lessons" <= CURRENT_DATE
8 AND "Lesson"."Date_of_lessons" > (CURRENT_DATE - 31)
9 AND "Class"."Class_type" = 'компьютерный класс'
10 GROUP BY "Lesson"."L_ID_class";
11
12
```

Data Output Messages Notifications

Successfully run. Total query runtime: 139 msec.  
1 rows affected.

Запрос с индексом

CREATE INDEX av\_workload ON "Lesson" ("L\_ID\_class");





Общее время выполнения запроса с индексом: 38 msec

## Удаление индексов

The screenshot displays the SQL Server Enterprise Manager interface. The 'Query' tab is active, showing a single query: `Drop INDEX av_workload;`. The query is executed across 12 lines. Below the query editor, the 'Messages' tab is selected, displaying the message: 'Successfully run. Total query runtime: 81 msec.'

Line	Text
1	Drop INDEX av_workload;
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Messages

Successfully run. Total query runtime: 81 msec.

Query

Query History

1

Drop INDEX small\_num\_students;

2

3

4

5

6

7

8

9

10

11

12

Data Output

Messages

Notifications

Successfully run. Total query runtime: 41 msec.

Query

Query History

1

DROP INDEX sv\_classes

2

3

4

Data Output

Messages

Explain ×

No

DROP INDEX

Query returned successfully in 43 msec.

## **Выводы**

При выполнении данной лабораторной работы были приобретены практические навыки по работе с таблицами в базе данных PostgreSQL, выполнением запросов на выборку и модификацию данных, представления и индексы.

Были реализованы запросы на вставку, удаление и изменение данных. Были выполнены запросы с использованием индексов и без использования индексов. Можно сделать вывод, что при использовании индексов запрос работает быстрее: общее время выполнения запроса без индекса больше, чем общее время выполнения запроса с индексом.

Таким образом, был получен опыт по работе и использованию программы pgAdmin 4.