

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

## **Лабораторная работа № 2**

**«Запросы на выборку и модификацию данных,  
представления и индексы в PostgreSQL»**

Выполнил: Евдокимов Владислав Борисович

Группа: К3242

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург  
2022

**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 1X, 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.

**Выполнение:**

- запросы к базе данных. Выполнить запросы согласно индивидуальному заданию, часть 2. В отчете привести формулировку запроса, команду, скриншот выполнения запроса;
- представления. Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3. В отчете привести формулировку запроса, команду, скриншот выполнения запроса и просмотр содержимого представления;

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

# ИНФОРМАЦИЯ О БАЗЕ:

## 1. Название БД

### Вариант 7. «Курсы»

Описание предметной области: Подразделение занимается организацией внебюджетного образования. Имеется несколько типов краткосрочных курсов, предназначенных для определенных специальностей, связанных с программным обеспечением ИТ. Каждый тип курсов имеет определенную длительность и свой перечень изучаемых дисциплин. На каждую программу может быть набрано несколько групп обучающихся. По каждой дисциплине могут проводиться лекционные и лабораторные занятия. 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.

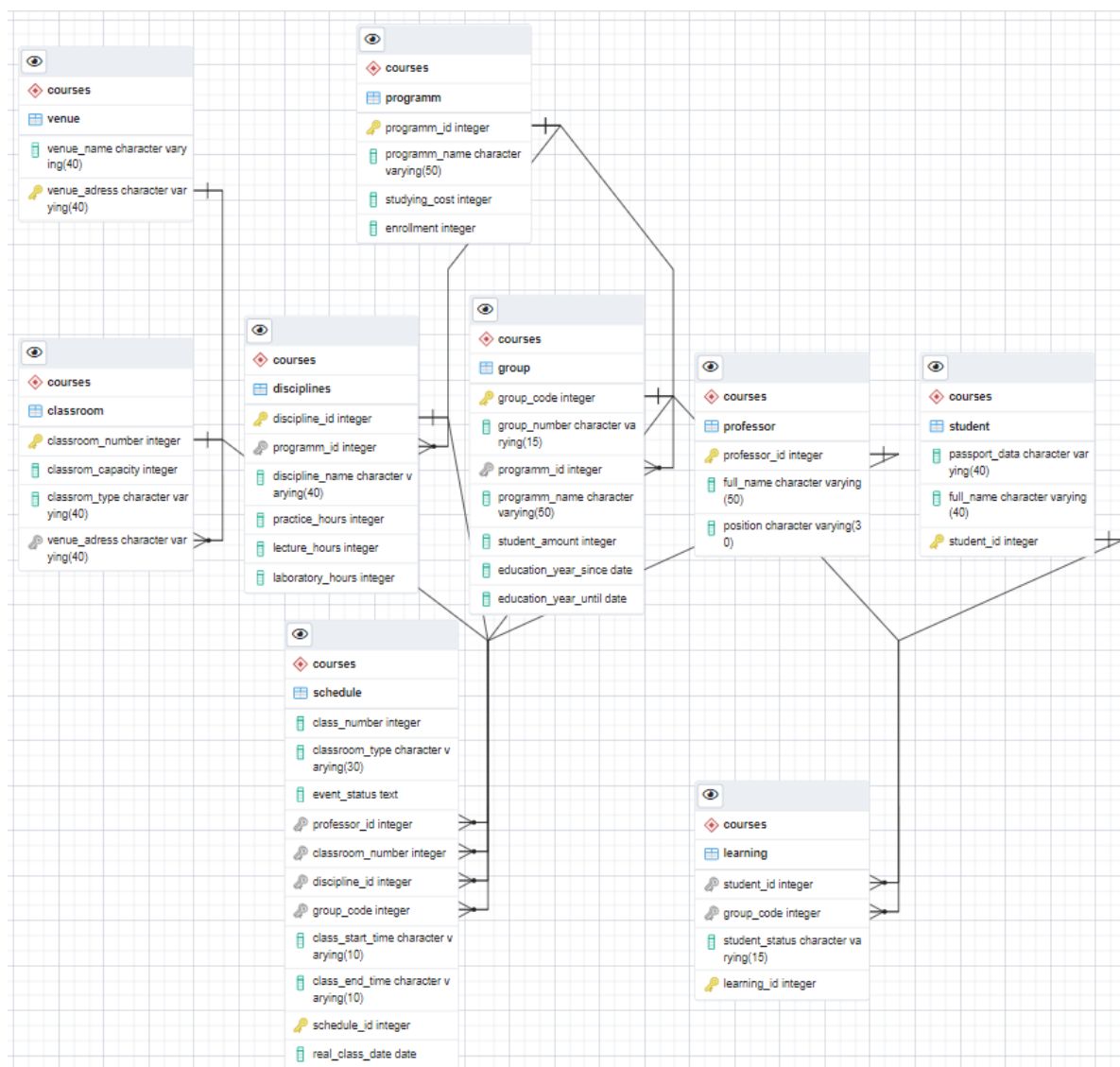
### Состав реквизитов сущностей:

- a) **Направление** (Код программы, наименование)
- b) **Дисциплины** (ID дисциплины, код направления, название дисциплины, лекционные часы, лабораторные часы, практические часы)
- c) **Группа** (Код группы, номер группы, год обучения, код направления)
- d) **Слушатель** (ID слушателя, контакты, имя, фамилия, код группы, паспортные данные)
- e) **Расписание** (Код расписания, ID преподавателя, ID дисциплины, код группы, тип занятий, номер пары, кол-во часов, номер аудитории, статус проведения, даты занятий, тип аудитории, номер класса, врем конца занятий, время начала занятий)
- f) **Площадка проведения** (Название, адрес)

g) **Аудитория** (Номер аудитории, тип аудитории, вместимость, название площадки)

h) **Преподаватель** (ID преподавателя, ФИО, должность)

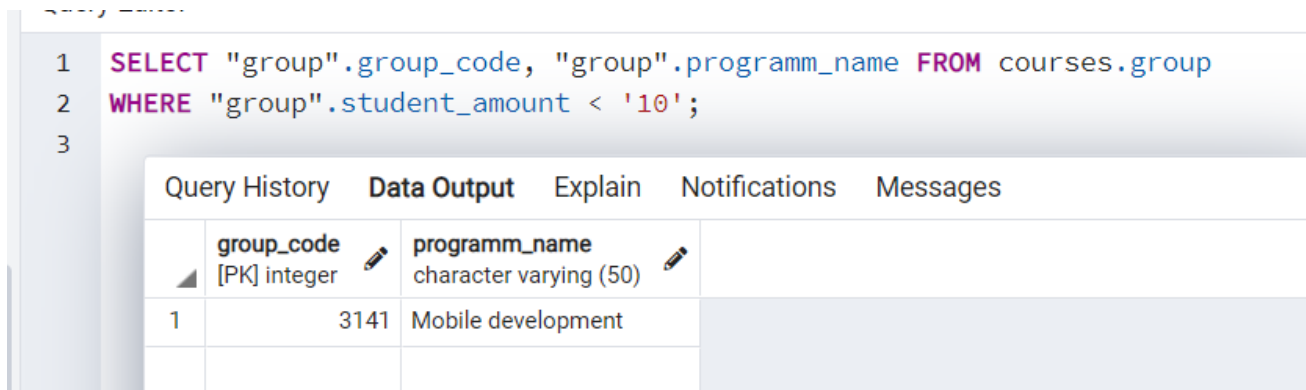
## 2. Схема логической модели базы данных:



## Запросы к базе данных

- 1) Вывести все номера групп и программы, где количество слушателей меньше 10:

```
SELECT "group".group_code, "group".programm_name FROM  
courses.group  
WHERE "group".student_amount < '10';
```



The screenshot shows a database query interface. On the left, the SQL query is displayed with line numbers 1, 2, and 3. The query is: `SELECT "group".group_code, "group".programm_name FROM courses.group WHERE "group".student_amount < '10';`. On the right, there is a tabbed interface with tabs for "Query History", "Data Output", "Explain", "Notifications", and "Messages". The "Data Output" tab is active, showing a table with two columns: "group\_code" (integer, PK) and "programm\_name" (character varying (50)). The table contains one row with the values "3141" and "Mobile development".

	group_code [PK] integer	programm_name character varying (50)
1	3141	Mobile development




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

```
SELECT courses.professor.full_name, COUNT(programm_id) AS  
programms_amount  
FROM courses.schedule  
NATURAL JOIN courses.group  
NATURAL JOIN courses.professor  
WHERE real_class_date > (NOW() - INTERVAL '1 YEAR') AND  
real_class_date < (NOW() - INTERVAL '6 months')  
GROUP BY full_name;
```

```

1 SELECT courses.professor.full_name, COUNT(programm_id) AS programms_amount
2 FROM courses.schedule
3 NATURAL JOIN courses.group
4 NATURAL JOIN courses.professor
5 WHERE real_class_date > (NOW() - INTERVAL '1 YEAR') AND real_class_date < (NOW() - INTERVAL '6 months')
6 GROUP BY full_name;
7

```

Messages		Notifications		Query History		Data Output		Explain	
	<b>full_name</b> character varying (50)		<b>programms_amount</b> bigint						
1	Govorova Marina Mikhaylova				1				
2	Milovanovich Ekaterina				1				
3	Sarah Daniali				2				
4	Sorokin Sergey				1				

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

```

SELECT DISTINCT schedule.professor_id FROM courses.schedule
WHERE schedule.professor_id NOT IN (SELECT
schedule.professor_id FROM
courses.schedule WHERE courses.schedule.class_number = '3');

```

Query Editor   Query History   Explain   Notifications

```

1 SELECT DISTINCT schedule.professor_id FROM courses.schedule
2 WHERE schedule.professor_id NOT IN (SELECT schedule.professor_id FROM
3 courses.schedule WHERE courses.schedule.class_number = '3');
4

```

Data Output

	professor_id integer
1	70

- 4) Вывести список свободных лекционных аудиторий на ближайший понедельник:

```

select distinct schedule.classroom_number from courses.schedule
where classroom_number not in (select classroom_number from
courses.schedule

```

where real\_class\_date = DATE(DATE\_TRUNC('week', NOW()) +  
INTERVAL '7 days') and schedule.classroom\_type = 'Lecture');

```
1 select distinct schedule.classroom_number from courses.schedule
2 where classroom_number not in (select classroom_number from courses.schedule
3 where real_class_date = DATE(DATE_TRUNC('week', NOW()) + INTERVAL '7 days') and schedule.classroom_type = 'Lecture');
```

	Messages	Notifications	Query History	Data Output	Explain
		classroom_number integer			
1		204			
2		3212			
3		206			
4		355			
5		2337			
6		103			
7		214			
8		1224			

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

SELECT "group".programm\_name, "group".student\_amount

FROM courses."group"

where "group".education\_year\_since BETWEEN (NOW() - INTERVAL '1  
YEAR') AND (SELECT CURRENT\_DATE)

GROUP BY "group".programm\_name, "group".student\_amount;

```
1 SELECT "group".programm_name, "group".student_amount
2 FROM courses."group"
3 where "group".education_year_since BETWEEN (NOW() - INTERVAL '1 YEAR') AND (SELECT CURRENT_DATE)
4 GROUP BY "group".programm_name, "group".student_amount;
```

	Messages	Notifications	Query History	Data Output	Explain
		programm_name character varying (50)		student_amount integer	
1		Intelligent Systems in Humanities		0	
2		Intelligent Systems in Humanities		20	
3		Mobile development		8	

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



```

SELECT classroom_number, ROUND(COUNT(*)*1.5/4, 3) AS
average_workload
FROM courses.schedule
WHERE real_class_date <= CURRENT_DATE
AND real_class_date > (CURRENT_DATE - 31)
AND classroom_type = 'Laboratory'
GROUP BY classroom_number;

```

#### Query Editor

```

1 SELECT classroom_number, ROUND(COUNT(*)*1.5/4, 3) AS average_workload
2 FROM courses.schedule
3 WHERE real_class_date <= CURRENT_DATE
4 AND real_class_date > (CURRENT_DATE - 31)
5 AND classroom_type = 'Laboratory'
6 GROUP BY classroom_number;

```

Query History Explain Notifications Data Output Messages

	classroom_number integer	average_workload numeric
1	204	0.750
2	214	0.375

7) Найти самые популярные программы за последние 3 года:

```

select programm.programm_name, programm.Enrollment,
programm.studying_cost, programm.programm_id
from courses.programm order by (programm.Enrollment) DESC

```

#### Query Editor

```

1 select programm.programm_name, programm.Enrollment, programm.studying_cost, programm.programm_id
2 from courses.programm order by (programm.Enrollment) DESC

```

Query History Explain Notifications Data Output Messages

	programm_name character varying (50)	enrollment integer	studying_cost integer	programm_id [PK] integer
1	Intelligent Systems in Humanities	54	208000	450345663
2	Mobile development	29	220000	4504068

## Графические представления:

- 1) Создать представление для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов:

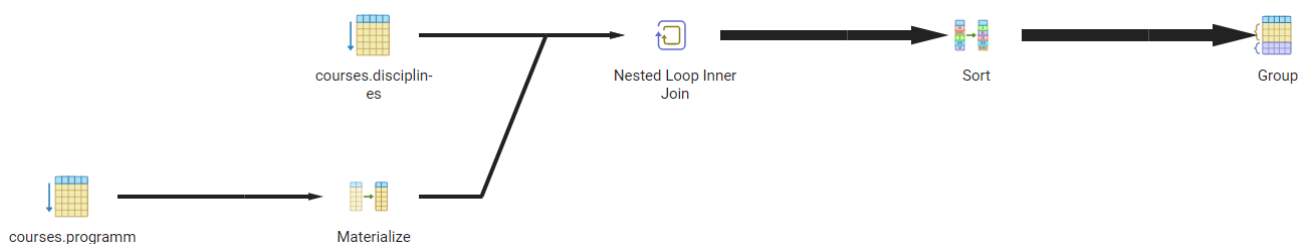
```
CREATE VIEW courses.All_Programms AS
SELECT courses.programm.programm_name,
courses.disciplines.discipline_name AS Disciplines,
(practice_hours + lecture_hours + laboratory_hours) as Total_hours
FROM courses.programm, courses.disciplines
GROUP BY courses.programm.programm_name,
courses.disciplines.discipline_name, Total_hours
ORDER BY courses.programm.programm_name,
courses.disciplines.discipline_name DESC;
```

```

1 CREATE VIEW courses.All_Programms AS
2 SELECT courses.programm.programm_name, courses.disciplines.discipline_name AS Disciplines,
3 (practice_hours + lecture_hours + laboratory_hours) as Total_hours
4 FROM courses.programm, courses.disciplines
5 GROUP BY courses.programm.programm_name, courses.disciplines.discipline_name, Total_hours
6 ORDER BY courses.programm.programm_name, courses.disciplines.discipline_name DESC;
7
8 select * from courses.All_Programms

```

	programm_name	disciplines	total_hours	
	character varying (50)	character varying (40)	integer	
6	Intelligent Systems in Humanities	Mathematical Analysis	143	
7	Intelligent Systems in Humanities	Machine learning	100	
8	Intelligent Systems in Humanities	Foreign language	108	
9	Intelligent Systems in Humanities	Economics	108	
10	Intelligent Systems in Humanities	Ecology	108	
11	Intelligent Systems in Humanities	Database creation	108	
12	Mobile development	Project Management	108	
13	Mobile development	Programming	95	
14	Mobile development	Probability theory	108	
15	Mobile development	Physics	108	
16	Mobile development	Physical education	40	



2) Создать представление, содержащее сведения об общем доходе по каждой программе за последний год:

```
CREATE VIEW courses.Income AS
```

```
SELECT programm.programm_name, programm.studying_cost AS cost
```

```
FROM courses.programm
```

```
GROUP BY programm_name, studying_cost
```

```
ORDER BY cost DESC;
```

```

1 CREATE VIEW courses.Income AS
2 SELECT programm.programm_name, programm.studying_cost AS cost
3 FROM courses.programm
4 GROUP BY programm_name, studying_cost
5 ORDER BY cost DESC;
6
7 select * from courses.Income

```

Data Output Query History Explain Notifications Messages

	programm_name character varying (50)	cost integer
1	Mobile development	220000
2	Intelligent Systems in Humanities	208000



## Запросы на модификацию данных:

- 1) **Insert** – Добавить группе 3141 в расписание занятие по дисциплине «Мобильная разработка» 2022-09-10 в 13:30.

# Запрос:

```
1 INSERT INTO courses.schedule(class_number, classroom_type, event_status,  
2 professor_id, classroom_number, discipline_id, group_code, class_start_time, class_end_time, schedule_id, real_class_date)  
3 VALUES(1, 'Lecture', 'In the Future',  
4 (SELECT professor.professor_id FROM courses.professor WHERE professor_id = '101'),  
5 (SELECT classroom.classroom_number FROM courses.classroom WHERE classrom_type= 'Lecture' LIMIT 1),  
6 (SELECT disciplines.discipline_id FROM courses.disciplines WHERE discipline_id= '18'),  
7 '3141', '13:30', '15:00', '157', '2020-09-10');  
8
```

Messages Notifications Query History Data Output Explain

INSERT 0 1

Query returned successfully in 193 msec.

# До:

22	3	Laboratory	Pass	74	204	88	3242	10:00	11:30	
23	1	Lecture	Pass	99	204	15	3141	8:20	9:50	
24	3	Lecture	Pass	73	1224	17	3141	8:20	9:50	
25	1	Laboratory	Pass	70	1224	345678	3141	8:20	9:50	
26	2	Lecture	Pass	60	2337	15	3141	8:20	9:50	

# После:

22	3	Laboratory	Pass	74	204	88	3242	10:00	11:30	7 20
23	1	Lecture	Pass	99	204	15	3141	8:20	9:50	23 20
24	3	Lecture	Pass	73	1224	17	3141	8:20	9:50	25 20
25	1	Laboratory	Pass	70	1224	345678	3141	8:20	9:50	26 20
26	2	Lecture	Pass	60	2337	15	3141	8:20	9:50	27 20
27	1	Lecture	In the Future	101	355	18	3141	13:30	15:00	157 20

## 2) **Delete** – Отменить занятие 2 по английскому языку на неделе для группы 3241:

### **Запрос:**

```
1 Delete from courses.schedule where schedule.schedule_id in
2 (select schedule_id from courses.schedule
3  where schedule.discipline_id = '8' and schedule.group_code = '3141');
```

Messages Notifications Query History Data Output Explain

DELETE 1

Query returned successfully in 100 msec.

### **До:**

	class_number integer	classroom_type character varying (30)	event_status text	professor_id integer	classroom_number integer	discipline_id integer	group_code integer	class_start_time character varying (10)	class_end_time character varying (10)	schedule_i [PK] integer
19		3 Lecture	Pass	60	206	17	3242	10:00	11:30	
20		3 Lecture	Pass	60	206	17	3242	10:00	11:30	
21		3 Lecture	Pass	60	206	17	3242	10:00	15:00	
22		3 Laboratory	Pass	74	204	88	3242	10:00	11:30	
23		1 Lecture	Pass	99	204	15	3141	8:20	9:50	
24		2 Lecture	Pass	73	103	8	3141	8:20	9:50	
25		3 Lecture	Pass	73	1224	17	3141	8:20	9:50	
26		1 Laboratory	Pass	70	1224	345678	3141	8:20	9:50	
27		2 Lecture	Pass	60	2337	15	3141	8:20	9:50	

### **После:**

	class_number integer	classroom_type character varying (30)	event_status text	professor_id integer	classroom_number integer	discipline_id integer	group_code integer	class_start_time character varying (10)	class_end_time character varying (10)	sch [PK]
19		3 Lecture	Pass	60	206	17	3242	10:00	11:30	
20		3 Lecture	Pass	60	206	17	3242	10:00	11:30	
21		3 Lecture	Pass	60	206	17	3242	10:00	15:00	
22		3 Laboratory	Pass	74	204	88	3242	10:00	11:30	
23		1 Lecture	Pass	99	204	15	3141	8:20	9:50	
24		3 Lecture	Pass	73	1224	17	3141	8:20	9:50	
25		1 Laboratory	Pass	70	1224	345678	3141	8:20	9:50	
26		2 Lecture	Pass	60	2337	15	3141	8:20	9:50	

3) **Update** – отчислить самых первых шести студентов образовательного учреждения по причине завершения обучения в нём:

**Запрос:**

```
1 UPDATE courses.learning SET student_status = 'Expelled' WHERE student_id IN
2 (SELECT learning.student_id from courses.learning where student_id < '7' )
```

Messages Notifications Query History Data Output Explain

UPDATE 6

Query returned successfully in 109 msec.

**До:**

1	1	3141	Student	111
2	2	3242	Student	222
3	3	3141	Student	333
4	4	3141	Student	444
5	5	3141	Student	555
6	6	3141	Student	666
7	7	3141	Student	777
8	8	3141	Student	888
9	9	3141	Student	999
10	11	3242	Student	1111
11	12	3242	Student	1212
12	13	3242	Student	1313

**После:**

22	28	3242	Student	2828
23	1	3141	Expelled	111
24	3	3141	Expelled	333
25	4	3141	Expelled	444
26	5	3141	Expelled	555
27	2	3242	Expelled	222
28	6	3242	Expelled	1010

## Создание индексов:

### 1) Запрос без индекса:

```

1 SELECT classroom_number, ROUND(COUNT(*)*1.5/4, 3) AS average_workload
2 FROM courses.schedule
3 WHERE real_class_date <= CURRENT_DATE
4 AND real_class_date > (CURRENT_DATE - 31)
5 AND classroom_type = 'Laboratory'
6 GROUP BY classroom_number;

```

7 Data Output Query History Explain Notifications Messages

Successfully run. Total query runtime: 141 msec.  
2 rows affected.



### Запрос с индексом:



```
1 CREATE INDEX average_workload ON courses.schedule (classroom_number)
```

Data Output Query History Explain Notifications Messages

CREATE INDEX

Query returned successfully in 123 msec.

Query Editor

```
1 SELECT classroom_number, ROUND(COUNT(*)*1.5/4, 3) AS average_workload
2 FROM courses.schedule
3 WHERE real_class_date <= CURRENT_DATE
4 AND real_class_date > (CURRENT_DATE - 31)
5 AND classroom_type = 'Laboratory'
6 GROUP BY classroom_number;
7
```

Query History Data Output Explain Notifications Messages

Successfully run. Total query runtime: 92 msec.  
2 rows affected.



## 2) ЗАПРОС БЕЗ ИНДЕКСА

Query Editor

```
1 SELECT "group".group_code, "group".programm_name FROM courses.group
2 WHERE "group".student_amount < '10';
3
```

Messages Notifications Query History Data Output Explain

Successfully run. Total query runtime: 194 msec.  
1 rows affected.

Создание индекса:

Query Editor

```
1 CREATE INDEX lesser_student_number ON courses."group" (group_code)
```

Messages Notifications Query History Data Output Explain

CREATE INDEX

Query returned successfully in 123 msec.

```
1 SELECT "group".group_code, "group".programm_name FROM courses.group
2 WHERE "group".student_amount < '10';
3
```

Messages Notifications Query History Data Output Explain

Successfully run. Total query runtime: 106 msec.  
1 rows affected.

### Удалить индексы:

```
1 drop INDEX courses.average_workload
```

Messages Notifications Query History Data Output Explain

DROP INDEX

Query returned successfully in 99 msec.

Query Editor

```
1 drop INDEX courses.lesser_student_number
```

Messages Notifications Query History Data Output

DROP INDEX

Query returned successfully in 80 msec.

## **Вывод:**

С помощью PgAdmin 4 я обуздал и создал комплексные запросы и представления информации в базе данных, запросы INSERT, DELETE, UPDATE, выяснил, что использование индексов помогает сократить время доступа к данным в таблицах.