Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №2

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

по дисциплине: «Проектирование и реализация базы данных»

Выполнил:
студентка III курса ИКТ
группы К33402
Самчук Анита Алексеевна

Проверил:
Говорова Марина Михайловна

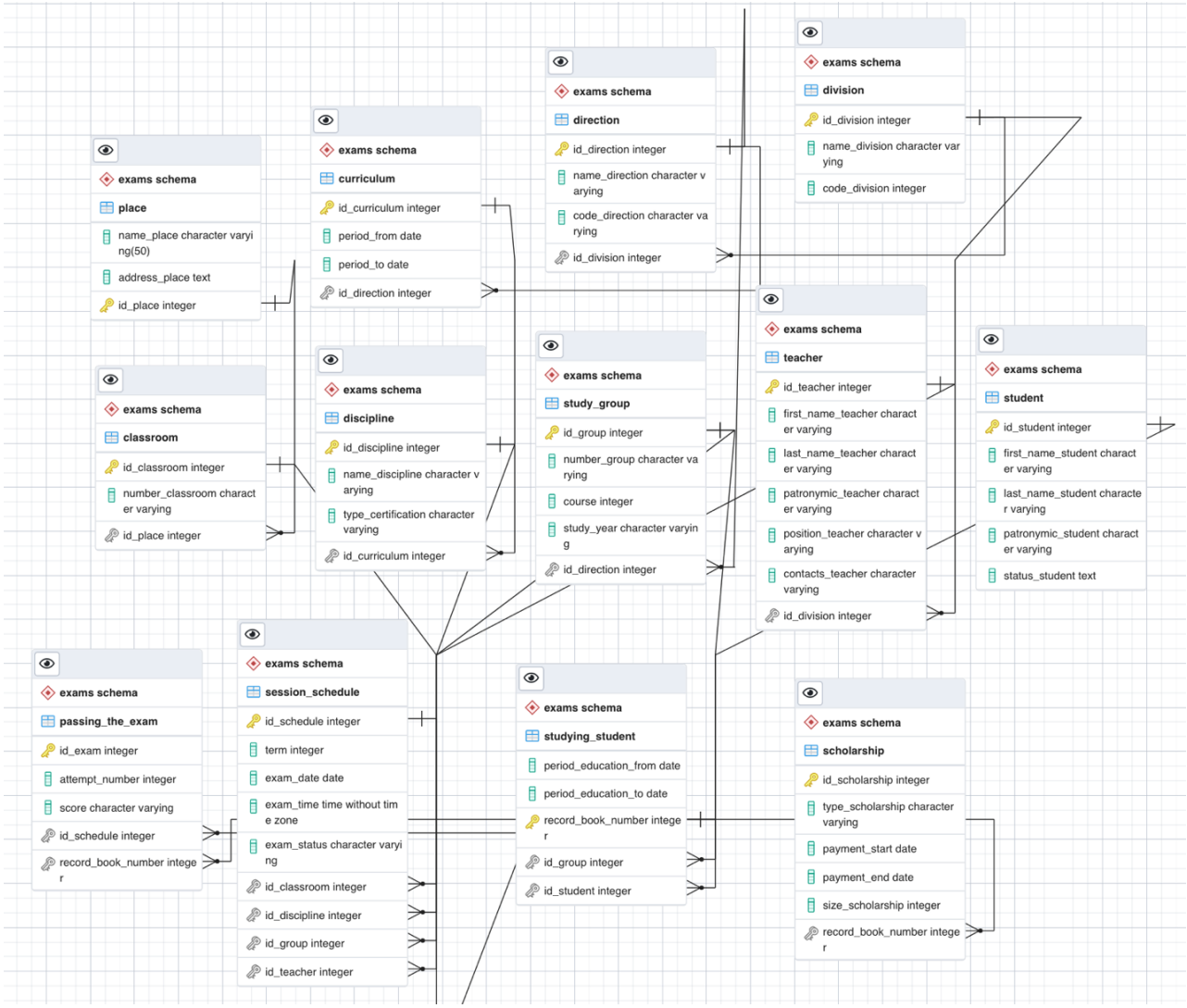
Санкт-Петербург
2020

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

Схема базы данных:



Ход работы:

Задние 1. Создайте запросы

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

```
select name_discipline, last_name_teacher, exam_date
from "exams schema".session_schedule
inner join "exams schema".discipline using(id_discipline)
inner join "exams schema".teacher using(id_teacher)
where id_group = 1
```

	name_discipline character varying	last_name_teacher character varying	exam_date date
1	Экология	Ватьян	2022-07-15
2	Проектирование и реализация баз данных	Говорова	2022-06-30

2. Вывести список студентов, получивших 0 баллов на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```
select last_name_student, first_name_student, last_name_teacher
from "exams schema".passing_the_exam
inner join "exams schema".session_schedule using (id_schedule)
inner join "exams schema".student using (id_student)
inner join "exams schema".teacher using(id_teacher)
where attempt_number = 1 and (mark = 'Не зачет' or mark = '2')
```

Query Editor | Query History

```
1  select last_name_student, first_name_student, last_name_teacher
2  from "exams schema".passing_the_exam
3  inner join "exams schema".session_schedule using (id_schedule)
4  inner join "exams schema".student using (id_student)
5  inner join "exams schema".teacher using(id_teacher)
6  where attempt_number = 1 and (mark = 'Не зачет' or mark = '2')
```

Data Output | Explain | Messages | Notifications

	last_name_student character varying	first_name_student character varying	last_name_teacher character varying
1	Казанский	Степан	Ватьян
2	Казанский	Степан	Говорова
3	Казанский	Степан	Хлопотов

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

```
select distinct last_name_student
from "exams schema".passing_the_exam
inner join "exams schema".student using(id_student)
where(mark not in ('Зачет', 'Незачет')) and
(cast(mark as int) >
        (select avg(cast(mark as int))
         from "exams schema".passing_the_exam
         inner join "exams schema".student using(id_student)
         inner join "exams schema".session_schedule using(id_schedule)
         inner join "exams schema".discipline using(id_discipline)
         where name_discipline = 'Проектирование и реализация баз данных'))
```

Query Editor | Query History

```
1  select distinct last_name_student
2  from "exams schema".passing_the_exam
3  inner join "exams schema".student using(id_student)
4  where(mark not in ('Зачет', 'Незачет')) and
5  (cast(mark as int) >
6   (select avg(cast(mark as int))
7    from "exams schema".passing_the_exam
8    inner join "exams schema".student using(id_student)
9    inner join "exams schema".session_schedule using(id_schedule)
10   inner join "exams schema".discipline using(id_discipline)
11   where name_discipline = 'Проектирование и реализация баз данных'))
```

Data Output | Explain | Messages | Notifications

	last_name_student character varying
1	Кукрякова
2	Самчук

4. Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убыванию

```
select name_direction, number_group, round(total,2)
from (select id_group, avg(cast(mark as int)) as total
        from "exams schema".passing_the_exam
        inner join "exams schema".session_schedule using(id_schedule)
        where(mark not in ('Зачет', 'Незачет'))
        group by id_group) as groups
inner join "exams schema".study_group using(id_group)
inner join "exams schema".direction using(id_direction)
where name_direction = 'Мобильные и сетевые технологии'
order by total desc
```

Query Editor | Query History

```
select name_direction, number_group, round(total,2)
from (select id_group, avg(cast(mark as int)) as total
      from "exams schema".passing_the_exam
      inner join "exams schema".session_schedule using(id_schedule)
      where(mark not in ('Зачет', 'Незачет'))
      group by id_group) as groups
inner join "exams schema".study_group using(id_group)
inner join "exams schema".direction using(id_direction)
where name_direction = 'Мобильные и сетевые технологии'
order by total desc
```

Data Output | Explain | Messages | Notifications

	name_direction character varying	number_group character varying	round numeric
1	Мобильные и сетевые технологии	K33401	4.00

5. Создайте списки студентов, упорядоченные по 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 distinct last_name_student, first_name_student, ROUND(average_mark,2), scholarship
from (select record_book_number,
avg(cast(mark as int)) as average_mark,
MIN(CASE WHEN mark > '3' AND mark != 'Незачет' THEN 1 ELSE 0 END) AS
scholarship
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as students
INNER JOIN "exams schema".studying_student USING (record_book_number)
INNER JOIN "exams schema".student USING (id_student)
ORDER BY last_name_student

Query Editor | Query History

```
select distinct last_name_student, first_name_student, ROUND(average_mark,2), scholarship
from (select record_book_number,
avg(cast(mark as int)) as average_mark,
MIN(CASE WHEN mark > '3' AND mark != 'Незачет' THEN 1 ELSE 0 END) AS
scholarship
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as students
INNER JOIN "exams schema".studying_student USING (record_book_number)
INNER JOIN "exams schema".student USING (id_student)
ORDER BY last_name_student
```

Data Output | Explain | Messages | Notifications

	last_name_student character varying	first_name_student character varying	round numeric	scholarship integer
1	Казанский	Степан	2.00	0
2	Кукрякова	Рената	5.00	1
3	Самчук	Анита	4.67	1

6. Вывести список студентов, сдавших все положенные экзамены

```
SELECT DISTINCT record_book_number, last_name_student, first_name_student
FROM (SELECT record_book_number,
SUM(CASE WHEN mark >= '3' AND mark != 'Незачет' THEN 1 ELSE 0 END) as sum_mark,
COUNT(DISTINCT passing_the_exam.id_schedule) as count_exams FROM "exams
schema".passing_the_exam
GROUP BY record_book_number) as marks
INNER JOIN "exams schema".studying_student USING (record_book_number)
INNER JOIN "exams schema".student USING (id_student)
WHERE sum_mark = count_exams
ORDER BY last_name_student
```

Query Editor | Query History

```
1  SELECT DISTINCT record_book_number, last_name_student, first_name_student
2  FROM (SELECT record_book_number,
3  SUM(CASE WHEN mark >= '3' AND mark != 'Незачет' THEN 1 ELSE 0 END) as sum_mark,
4  COUNT(DISTINCT passing_the_exam.id_schedule) as count_exams FROM "exams schema".passing_the_exam
5  GROUP BY record_book_number) as marks
6  INNER JOIN "exams schema".studying_student USING (record_book_number)
7  INNER JOIN "exams schema".student USING (id_student)
8  WHERE sum_mark = count_exams
9  ORDER BY last_name_student
```

Data Output | Explain | Messages | Notifications

	record_book_number integer	last_name_student character varying	first_name_student character varying
1	1	Береснев	Андрей
2	8	Вдовенко	Мария
3	4	Кукрякова	Рената
4	5	Самчук	Анита

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

```
SELECT last_name_student, studying_student.id_group, ROUND(average_mark,2)
FROM (SELECT record_book_number, AVG(CAST(mark AS int)) as average_mark
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as all_students
INNER JOIN "exams schema".studying_student USING (record_book_number)
INNER JOIN "exams schema".student USING (id_student)
WHERE (studying_student.id_group, average_mark) = ANY(SELECT
studying_student.id_group, MAX(average_mark)
FROM (SELECT record_book_number, AVG(CAST(mark AS int)) as average_mark
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as all_students
INNER JOIN "exams schema".studying_student USING (record_book_number) GROUP BY
id_group)
ORDER BY studying_student.id_group, last_name_student
```

Query Editor | Query History

```
SELECT last_name_student, studying_student.id_group, ROUND(average_mark,2)
FROM (SELECT record_book_number, AVG(CAST(mark AS int)) as average_mark
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as all_students
INNER JOIN "exams schema".studying_student USING (record_book_number)
INNER JOIN "exams schema".student USING (id_student)
WHERE (studying_student.id_group, average_mark) = ANY(SELECT studying_student.id_group, MAX(average_mark)
FROM (SELECT record_book_number, AVG(CAST(mark AS int)) as average_mark
FROM "exams schema".passing_the_exam
WHERE mark NOT IN ('Зачет', 'Незачет') GROUP BY record_book_number) as all_students
INNER JOIN "exams schema".studying_student USING (record_book_number) GROUP BY id_group)
ORDER BY studying_student.id_group, last_name_student
```

Data Output | Explain | Messages | Notifications

	last_name_student character varying	id_group integer	round numeric
1	Самчук	1	4.67
2	Кукрякова	2	5.00
3	Казанский	3	2.00

Задание 2. Создайте представления

1. Список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен

```
create view "exams schema".faild_attempt as
select "exams schema".passing_the_exam.record_book_number as record_book_number,
"exams schema".passing_the_exam.mark as mark,
"exams schema".student.last_name_student as last_name_student,
"exams schema".teacher.last_name_teacher as last_name_teacher
from "exams schema".passing_the_exam
inner join "exams schema".session_schedule using (id_schedule)
inner join "exams schema".student using (id_student)
inner join "exams schema".teacher using(id_teacher)
where attempt_number = 1 and mark = '2';

select * from "exams schema".faild_attempt;
```

Query Editor | Query History

```
1  create view "exams schema".faild_attempt as
2  select "exams schema".passing_the_exam.record_book_number as record_book_number,
3  "exams schema".passing_the_exam.mark as mark,
4  "exams schema".student.last_name_student as last_name_student,
5  "exams schema".teacher.last_name_teacher as last_name_teacher
6  from "exams schema".passing_the_exam
7  inner join "exams schema".session_schedule using (id_schedule)
8  inner join "exams schema".student using (id_student)
9  inner join "exams schema".teacher using(id_teacher)
10 where attempt_number = 1 and mark = '2';
11
12 select * from "exams schema".faild_attempt;
```

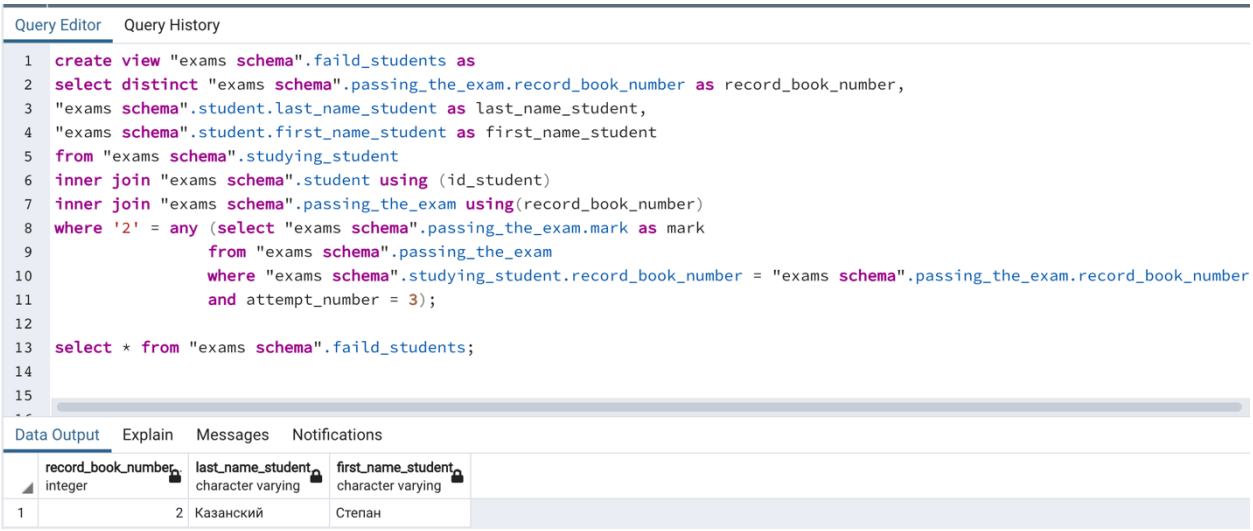
Data Output | Explain | Messages | Notifications

	record_book_number integer	mark character varying	last_name_student character varying	last_name_teacher character varying
1	2	2	Казанский	Ватьян
2	2	2	Казанский	Говорова

2. Данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки)

```
create view "exams schema".faild_students as
select distinct "exams schema".passing_the_exam.record_book_number as
record_book_number,
"exams schema".student.last_name_student as last_name_student,
"exams schema".student.first_name_student as first_name_student
from "exams schema".studying_student
inner join "exams schema".student using (id_student)
inner join "exams schema".passing_the_exam using(record_book_number)
where '2' = any (select "exams schema".passing_the_exam.mark as mark
                                    from "exams schema".passing_the_exam
                                    where "exams schema".studying_student.record_book_number =
"exams schema".passing_the_exam.record_book_number
                                    and attempt_number = 3);

select * from "exams schema".faild_students;
```



Задание 3: Создайте три запроса на модификацию данных

1. UPDATE (повышение стипендии отличникам на 10%)

До:

Query Editor | Query History

```
select * from "exams schema".scholarship
ORDER BY id_scholarship ASC;
```

Data Output | Explain | Messages | Notifications

	id_scholarship [PK] integer	type_scholarship character varying	payment_start date	payment_end date	size_scholarship integer	id_student integer
1	1	ГАС отлично успевающего студента	2022-09-01	2022-12-31	4100	313264
2	2	ГАС	2022-09-01	2022-12-31	2000	309670
3	3	ПГАС	2022-09-01	2022-12-31	10000	264920
4	4	ГСС	2022-09-01	2022-12-31	3000	308535

update "exams schema".scholarship
set size_scholarship = size_scholarship + size_scholarship * 0.1 where id_scholarship in
(select id_scholarship from "exams schema".scholarship
where type_scholarship = 'ГАС отлично успевающего студента' and
current_timestamp between payment_start and payment_end);

select * from "exams schema".scholarship
ORDER BY id_scholarship ASC;

После:

```
update "exams schema".scholarship
set size_scholarship = size_scholarship + size_scholarship * 0.1 where id_scholarship in
(select id_scholarship from "exams schema".scholarship where type_scholarship = 'ГАС отлично успевающего студента' and
current_timestamp between payment_start and payment_end);

select * from "exams schema".scholarship
ORDER BY id_scholarship ASC;
```

Data Output | Explain | Messages | Notifications

	id_scholarship [PK] integer	type_scholarship character varying	payment_start date	payment_end date	size_scholarship integer	id_student integer
1	1	ГАС отлично успевающего студента	2022-09-01	2022-12-31	4510	313264
2	2	ГАС	2022-09-01	2022-12-31	2000	309670
3	3	ПГАС	2022-09-01	2022-12-31	10000	264920
4	4	ГСС	2022-09-01	2022-12-31	3000	308535

2. INSERT (добавление сдачи зачетafrom по Иностранному языку для студента 308535)

До:

```
1  SELECT * FROM "exams schema".passing_the_exam
2  ORDER BY id_exam ASC
```

Data Output | Explain

	id_exam [PK] integer	attempt_number integer	id_schedule integer	id_student integer	points integer	mark character varying	record_book_number integer
1	1	1	1	313264	20	5	5
2	2	1	2	309670	20	5	4
3	3	1	4	313264	20	5	5
4	4	1	3	282928	0	Незачет	2
5	5	1	5	264920	16	Зачет	1
6	6	1	6	247891	12	Зачет	8
7	7	1	4	282928	0	2	2
8	8	1	2	313264	18	4	5
9	9	1	3	264920	16	Зачет	1
10	10	1	1	309670	20	5	4
11	11	1	1	282928	0	2	2
12	12	3	7	282928	0	2	2

```
insert into "exams schema".passing_the_exam(
id_exam, attempt_number, id_schedule, id_student, points, mark, record_book_number)
values (13, 1, (select id_schedule from "exams schema".session_schedule
where id_group = 3
    and id_discipline = (select id_discipline from "exams schema".discipline
    where name_discipline = 'Иностранный язык')), 308535, 20, 'Зачет',
    (select record_book_number from "exams schema".studying_student
      where id_student = 308535));
```

После:

Query Editor | Query History

```
insert into "exams schema".passing_the_exam(
id_exam, attempt_number, id_schedule, id_student, points, mark, record_book_number)
values (13, 1, (select id_schedule from "exams schema".session_schedule
where id_group = 3
                and id_discipline = (select id_discipline from "exams schema".discipline
                                     where name_discipline = 'Иностранный язык')), 308535, 20, 'Зачет',
        (select record_book_number from "exams schema".studying_student
       where id_student = 308535));

select * from "exams schema".passing_the_exam
ORDER BY id_exam ASC;
```

Data Output | Explain | Messages | Notifications

	id_exam [PK] integer	attempt_number integer	id_schedule integer	id_student integer	points integer	mark character varying	record_book_number integer
1	1	1	1	313264	20	5	5
2	2	1	2	309670	20	5	4
3	3	1	4	313264	20	5	5
4	4	1	3	282928	0	Незачет	2
5	5	1	5	264920	16	Зачет	1
6	6	1	6	247891	12	Зачет	8
7	7	1	4	282928	0	2	2
8	8	1	2	313264	18	4	5
9	9	1	3	264920	16	Зачет	1
10	10	1	1	309670	20	5	4
11	11	1	1	282928	0	2	2
12	12	3	7	282928	0	2	2
13	13	1	3	308535	20	Зачет	3

3. DELETE (удаление всех аудитории, находящихся по адресу Биржевая линия, д.14)

До:

	id_classroom [PK] integer	number_classroom character varying	id_place integer
1	1	100	1
2	2	101	1
3	3	102	1
4	4	100	2
5	5	101	2
6	6	102	2
7	7	100	2
8	8	101	3
9	9	102	3

```
 delete from "exams schema".classroom
where id_place in (select id_place from "exams schema".place
                                   where address_place = 'Биржевая линия, д.14');
select * from "exams schema".classroom;
```

После:

Query Editor | Query History

```
1  delete from "exams schema".classroom
2  where id_place in (select id_place from "exams schema".place
3                     where address_place = 'Биржевая линия, д.14');
4
5  select * from "exams schema".classroom;
```

Data Output | Explain | Messages | Notifications

	id_classroom [PK] integer	number_classroom character varying	id_place integer
1	1	100	1
2	2	101	1
3	3	102	1
4	4	100	2
5	5	101	2
6	6	102	2
7	7	100	2

Задание 4: создать простой и составной индексы

1. Простой индекс

Query Editor | Query History

```
1  select * from "exams schema".session_schedule
2  where id_teacher = 105760
3
```

Data Output | Explain | Messages | Notifications

	id_schedule [PK] integer	term integer	exam_date date	exam_time time without time zone	exam_status character varying	id_classroom integer	id_discipline integer	id_group integer	id_teacher integer
1	1	1	2022-06-30	11:40:00	Дистанционно	1	1	1	105760
2	5	1	2022-05-27	17:00:00	Дистанционно	5	5	5	105760
3	7	3	2023-10-15	11:40:00	Очно	7	1	4	105760

19.09.2022 15:49:56 — Date
3 — Rows Affected
74 msec — Duration

Copy | Copy to Query Editor

```
select * from "exams schema".session_schedule
where id_teacher = 105760
```

Messages

```
Successfully run. Total query runtime: 74 msec.
3 rows affected.
```

Создание индекса:

Query Editor | Query History

```
1  create index govorova
2  on "exams schema".session_schedule(id_teacher)
3  where id_teacher = 105760
```

Data Output | Explain | Messages | Notifications

```
CREATE INDEX

Query returned successfully in 74 msec.
```

19.09.2022 15:54:12	3	69 msec
Date	Rows Affected	Duration

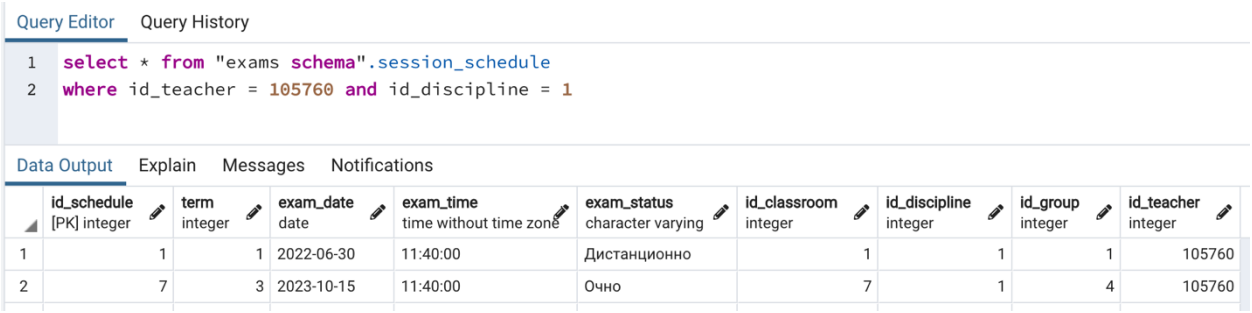
Copy | Copy to Query Editor

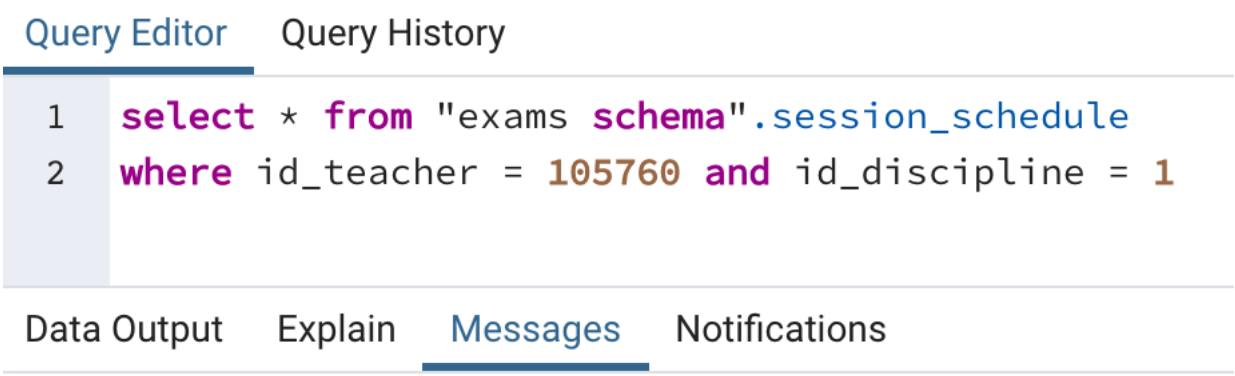
```
select * from "exams schema".session_schedule
where id_teacher = 105760
```

Messages

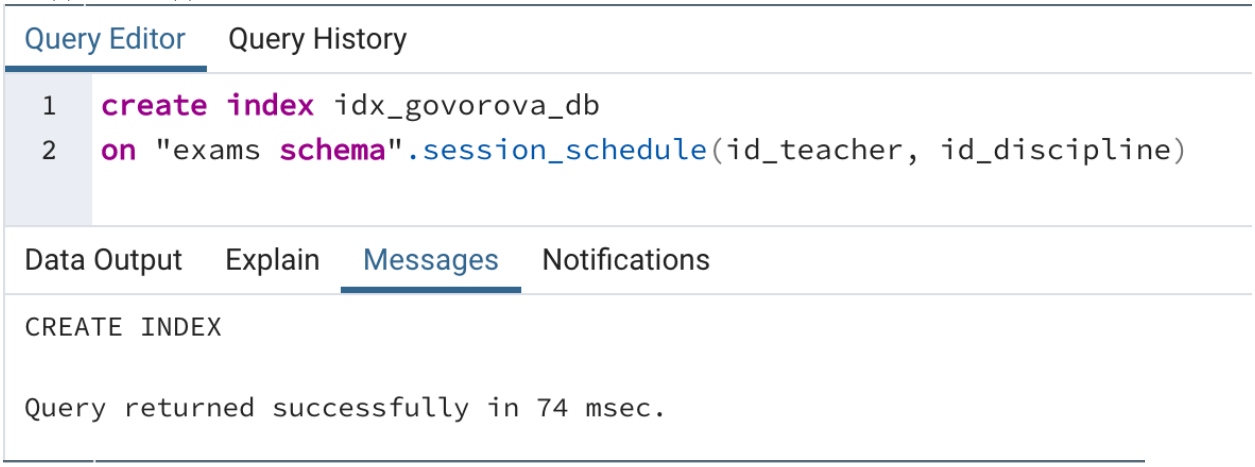
```
Successfully run. Total query runtime: 69 msec.
3 rows affected.
```

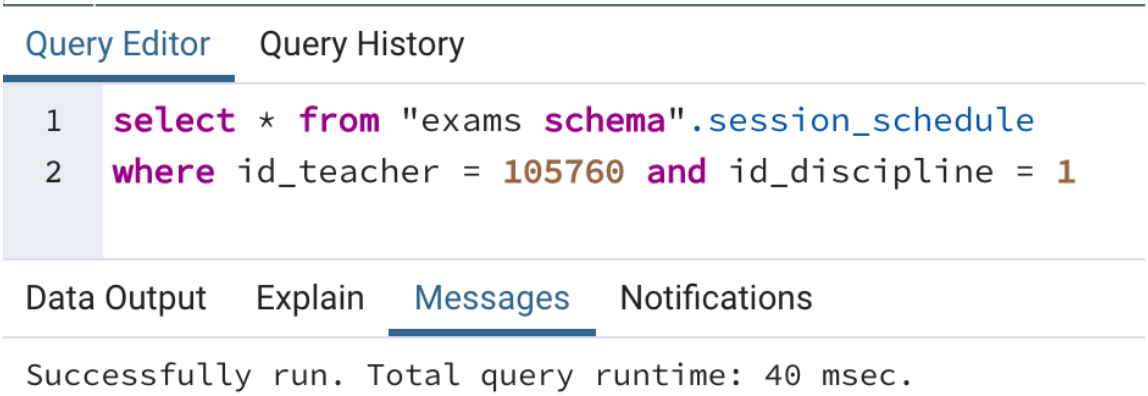
2. Составной индекс





Создание индекса:





Вывод: я овладела практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.