

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»**

Факультет инфокоммуникационных технологий

Дисциплина:

«Проектирование и реализация баз данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRE SQL»**

Выполнил:

студент группы К32391

Микитчак Иван Михайлович

(подпись)

Проверил:

Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

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.

Выполнение

Запросы к базе данных

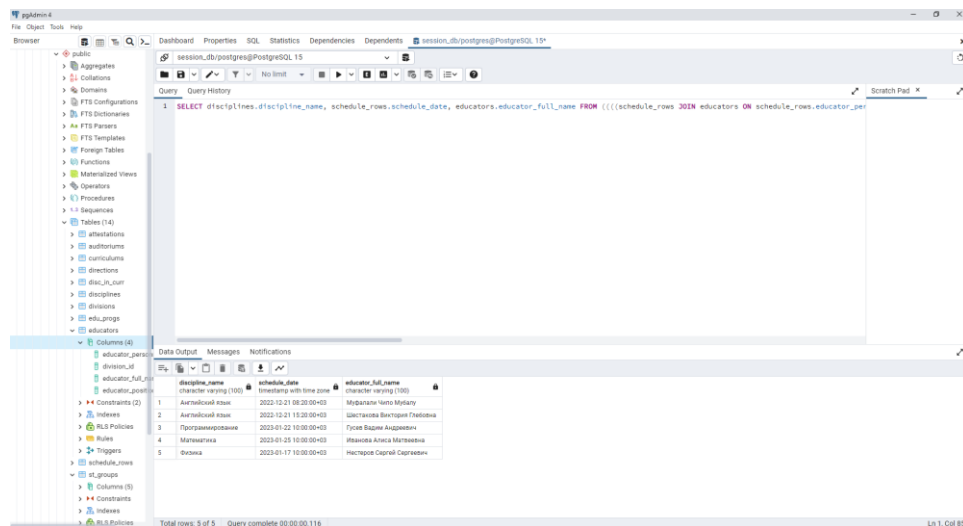
1

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

Команда:

```
SELECT disciplines.discipline_name,  
        schedule_rows.schedule_date,  
        educators.educator_full_name  
FROM (((schedule_rows  
        JOIN educators ON  
schedule_rows.educator_personnel_number =  
educators.educator_personnel_number)  
        JOIN st_groups ON schedule_rows.st_group_id =  
st_groups.st_group_id)  
        JOIN disc_in_curr ON schedule_rows.disc_in_curr_id =  
disc_in_curr.disc_in_curr_id)  
        JOIN disciplines ON disc_in_curr.discipline_id =  
disciplines.discipline_id)  
WHERE st_groups.st_group_number = 'K32391';
```

Результат:



The screenshot shows the pgAdmin 4 interface. The SQL query is entered in the Query tool, and the results are displayed in the Data Output pane. The results table has 5 rows and 4 columns: discipline_name, schedule_date, educator_full_name, and educator_personnel_number. The data is as follows:

discipline_name	schedule_date	educator_full_name	educator_personnel_number
Автоматизация	2022-12-21 08:28:00+03	Михайлова Елена Михайловна	1009
Автоматизация	2022-12-21 13:28:00+03	Щапова Елена Владимировна	1100
Программирование	2023-01-22 10:00:00+03	Гусев Евгений Александрович	1101
Математика	2023-01-25 10:00:00+03	Иванова Елена Александровна	1102
Физика	2023-01-17 10:00:00+03	Насредов Сергей Сергеевич	1103

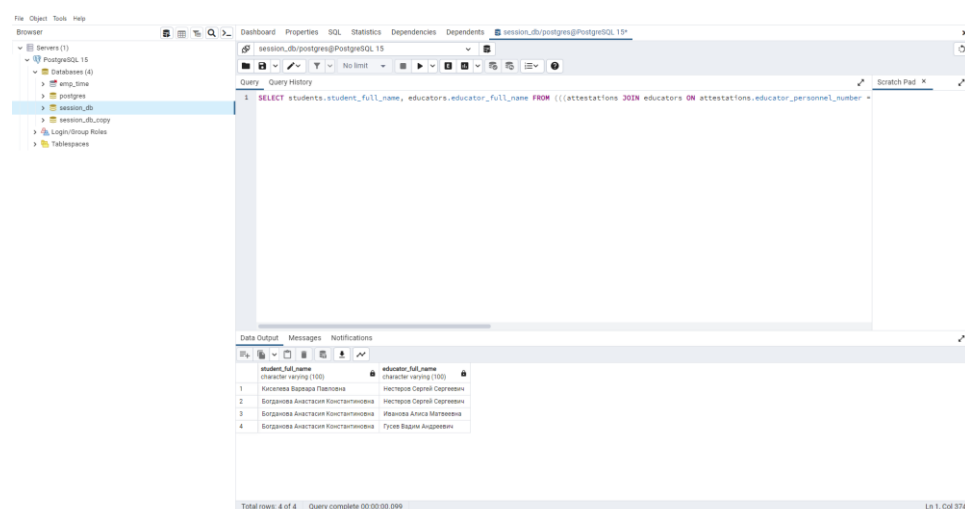
2

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

Команда:

```
SELECT students.student_full_name,  
       educators.educator_full_name  
FROM (((attestations  
       JOIN educators ON  
attestations.educator_personnel_number =  
educators.educator_personnel_number)  
       JOIN st_in_group ON  
attestations.student_personnel_number =  
st_in_group.student_personnel_number)  
       JOIN students ON st_in_group.student_id =  
students.student_id)  
WHERE attestations.mark = '2F'  
      AND attestations.attempt = 1;
```

Результат:



The screenshot shows a PostgreSQL query editor with the following SQL query:

```
SELECT students.student_full_name, educators.educator_full_name FROM (((attestations JOIN educators ON attestations.educator_personnel_number = educators.educator_personnel_number) JOIN st_in_group ON attestations.student_personnel_number = st_in_group.student_personnel_number) JOIN students ON st_in_group.student_id = students.student_id) WHERE attestations.mark = '2F' AND attestations.attempt = 1;
```

The query results are displayed in a table with 4 rows and 2 columns: student_full_name and educator_full_name.

	student_full_name	educator_full_name
1	Богданова Анастасия Константиновна	Иванов Сергей Сергеевич
2	Богданова Анастасия Константиновна	Иванова Анна Сергеевна
3	Богданова Анастасия Константиновна	Иванова Анна Сергеевна
4	Богданова Анастасия Константиновна	Гусев Вадим Андреевич

Total rows: 4 of 4. Query complete 00:00:00.099. Ln 1, Col 374.

3

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

Команда:

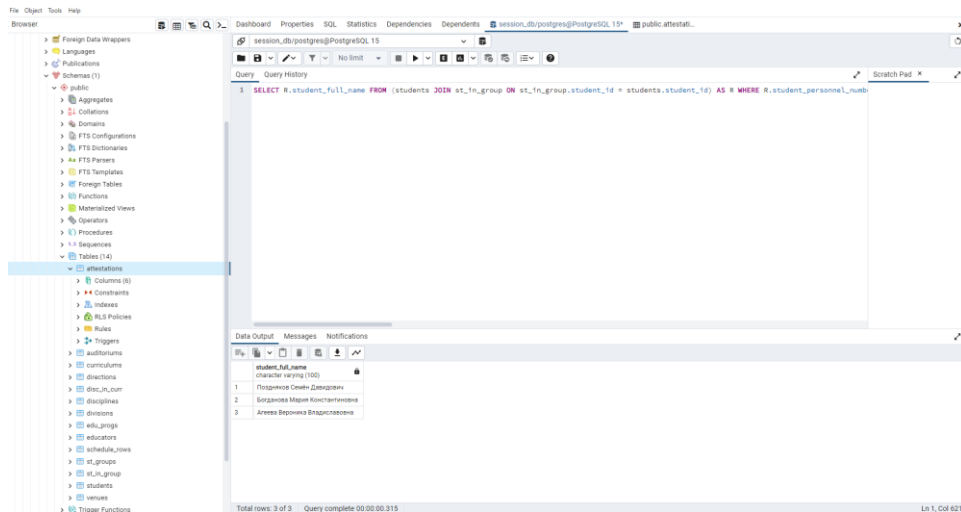
```
SELECT R.student_full_name  
FROM (students  
      JOIN st_in_group ON st_in_group.student_id =
```

```

students.student_id) AS R
WHERE R.student_personnel_number IN
    (SELECT T.student_personnel_number
     FROM
        (SELECT *,
             CASE
                WHEN mark='2F' THEN 2
                WHEN mark='3D' THEN 3
                WHEN mark='4C' THEN 4
                WHEN mark='4B' THEN 4
                WHEN mark='5A' THEN 5
                ELSE NULL
             END mark_to_num
         FROM attestations) AS T
     WHERE T.disc_in_curr_id = 2
        AND T.mark_to_num >
        (SELECT AVG(mark_to_num)
         FROM
            (SELECT *,
                 CASE
                    WHEN mark='2F' THEN 2
                    WHEN mark='3D' THEN 3
                    WHEN mark='4C' THEN 4
                    WHEN mark='4B' THEN 4
                    WHEN mark='5A' THEN 5
                    ELSE NULL
                 END mark_to_num
                FROM attestations) AS T
            WHERE disc_in_curr_id = 2))

```

Результат:



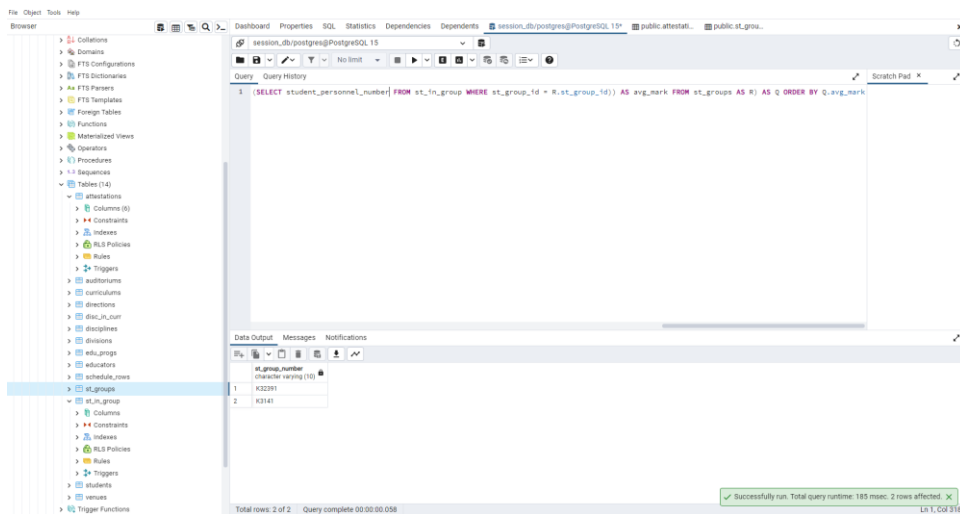
4

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

Команда:

```
SELECT Q.st_group_number
FROM
    (SELECT *,
        (SELECT AVG(T.mark_to_num)
         FROM
             (SELECT *,
                 CASE
                     WHEN mark='2F' THEN 2
                     WHEN mark='3D' THEN 3
                     WHEN mark='4C' THEN 4
                     WHEN mark='4B' THEN 4
                     WHEN mark='5A' THEN 5
                     ELSE NULL
                 END mark_to_num
              FROM attestations) AS T
            WHERE T.student_personnel_number IN
                (SELECT student_personnel_number
                 FROM st_in_group
                 WHERE st_group_id = R.st_group_id)) AS avg_mark
         FROM st_groups AS R) AS Q
    ORDER BY Q.avg_mark
```

Результат:



5

Формулировка: 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.

Команда:

```
SELECT V.st_group_number,
       V.student_full_name,
       V.avg,
       CASE
           WHEN V.sum = 0 THEN 1
           ELSE 0
       END scholarship
FROM (((st_in_group
        JOIN students ON st_in_group.student_id =
students.student_id)
      JOIN st_groups ON st_in_group.st_group_id =
st_groups.st_group_id)
     JOIN
      (SELECT T.student_personnel_number,
              SUM(T.f)
      FROM
        (SELECT *,
              CASE
                  WHEN mark='2F' THEN 1
```

```

        WHEN mark='3D' THEN 1
        WHEN mark='4C' THEN 0
        WHEN mark='4B' THEN 0
        WHEN mark='5A' THEN 0
        WHEN mark='Незачёт' THEN 1
        WHEN mark='Зачёт' THEN 0
        ELSE NULL
    END f
    FROM attestations) AS T
    GROUP BY T.student_personnel_number) AS R ON
st_in_group.student_personnel_number =
R.student_personnel_number)
JOIN
    (SELECT Q.student_personnel_number,
        AVG(Q.mark_to_num)
    FROM
        (SELECT *,
            CASE
                WHEN mark='2F' THEN 2
                WHEN mark='3D' THEN 3
                WHEN mark='4C' THEN 4
                WHEN mark='4B' THEN 4
                WHEN mark='5A' THEN 5
                ELSE NULL
            END mark_to_num
        FROM attestations) AS Q
    GROUP BY Q.student_personnel_number) AS P ON
st_in_group.student_personnel_number =
P.student_personnel_number) AS V
ORDER BY V.st_group_number,
    V.student_full_name

```

Результат:

The screenshot shows a PostgreSQL database interface. The left sidebar displays a tree view of the database schema, including tables, views, and sequences. The main window shows a SQL query being executed. The query is:

```
1. personnel_number) AS P ON st_in_group.student_personnel_number = P.student_personnel_number) AS Y ORDER BY Y.st_group_number, Y.student_full_name
```

The results are displayed in a table with the following columns: st_group_number, student_full_name, eng, and scholarship. The table contains 6 rows of data.

st_group_number	student_full_name	eng	scholarship
1	Алексеев Александр Владимирович	5.0000000000000000	1
2	Борисов Александр Константинович	2.0000000000000000	0
3	Борисов Мария Константиновна	5.0000000000000000	1
4	Колесова Валерия Павловна	3.3333333333333333	0
5	Прохорова Ольга Александровна	4.0000000000000000	1
6	Федосеев Алексей Александрович	4.3333333333333333	1

The status bar at the bottom indicates "Total rows: 6 of 6" and "Query complete 00:00:00.110".

6

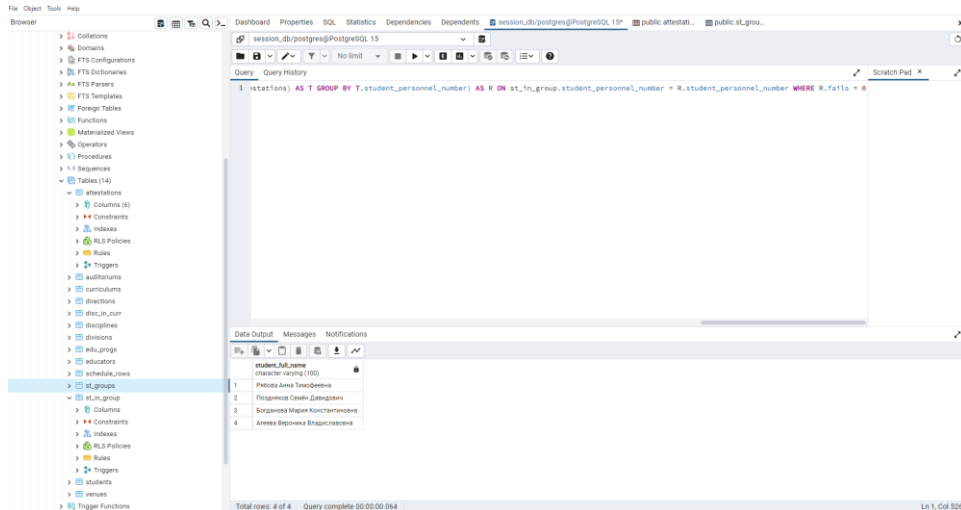
Формулировка: Вывести список студентов, сдавших все положенные экзамены.

Команда:

```
SELECT students.student_full_name
FROM (st_in_group
      JOIN students ON st_in_group.student_id =
students.student_id)
JOIN
  (SELECT T.student_personnel_number,
        SUM(T.f) AS fails
  FROM
    (SELECT *,
      CASE
        WHEN mark='2F' THEN 1
        WHEN mark='3D' THEN 1
        WHEN mark='4C' THEN 0
        WHEN mark='4B' THEN 0
        WHEN mark='5A' THEN 0
        WHEN mark='Незачёт' THEN 1
        WHEN mark='Зачёт' THEN 0
        ELSE NULL
      END f
    FROM attestations) AS T
  GROUP BY T.student_personnel_number) AS R ON
st_in_group.student_personnel_number =
```

```
R.student_personnel_number
WHERE R.fails = 0
```

Результат:



7

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

Команда:

```
SELECT V.student_full_name
FROM
    (SELECT st_groups.st_group_number,
            students.student_full_name,
            P.avg_mark
    FROM ((st_in_group
            JOIN students ON st_in_group.student_id =
students.student_id)
            JOIN st_groups ON st_in_group.st_group_id =
st_groups.st_group_id)
    JOIN
        (SELECT Q.student_personnel_number,
            AVG(Q.mark_to_num) AS avg_mark
        FROM
            (SELECT *,
                CASE
                    WHEN mark='2F' THEN 2
                    WHEN mark='3D' THEN 3
```

```

        WHEN mark='4C' THEN 4
        WHEN mark='4B' THEN 4
        WHEN mark='5A' THEN 5
        ELSE NULL
    END mark_to_num
FROM attestations) AS Q
GROUP BY Q.student_personnel_number) AS P ON
st_in_group.student_personnel_number =
P.student_personnel_number) AS V,

(SELECT V.st_group_number,
      MAX(V.avg_mark)
FROM
  (SELECT st_groups.st_group_number,
        students.student_full_name,
        P.avg_mark
  FROM ((st_in_group
        JOIN students ON st_in_group.student_id =
students.student_id)
        JOIN st_groups ON st_in_group.st_group_id =
st_groups.st_group_id)
  JOIN
    (SELECT Q.student_personnel_number,
          AVG(Q.mark_to_num) AS avg_mark
    FROM
      (SELECT *,
            CASE
              WHEN mark='2F' THEN 2
              WHEN mark='3D' THEN 3
              WHEN mark='4C' THEN 4
              WHEN mark='4B' THEN 4
              WHEN mark='5A' THEN 5
              ELSE NULL
            END mark_to_num
          FROM attestations) AS Q
      GROUP BY Q.student_personnel_number) AS P ON
st_in_group.student_personnel_number =
P.student_personnel_number) AS V
    GROUP BY V.st_group_number) AS U

```

WHERE V.st_group_number = U.st_group_number
AND V.avg_mark = U.max

Результат:

The screenshot shows a PostgreSQL database management tool interface. The left sidebar displays a tree view of database objects, with 'st_group' selected under the 'public' schema. The main window shows a SQL query in the 'Query' tab:

```
SELECT (l.number * P.student_personnel_number) AS V GROUP BY V.st_group_number AS U WHERE V.st_group_number = U.st_group_number AND V.avg_mark = U.max
```

The 'Data Output' tab shows the results of the query:

student_id_name
1
2

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.060'.

Создание представлений

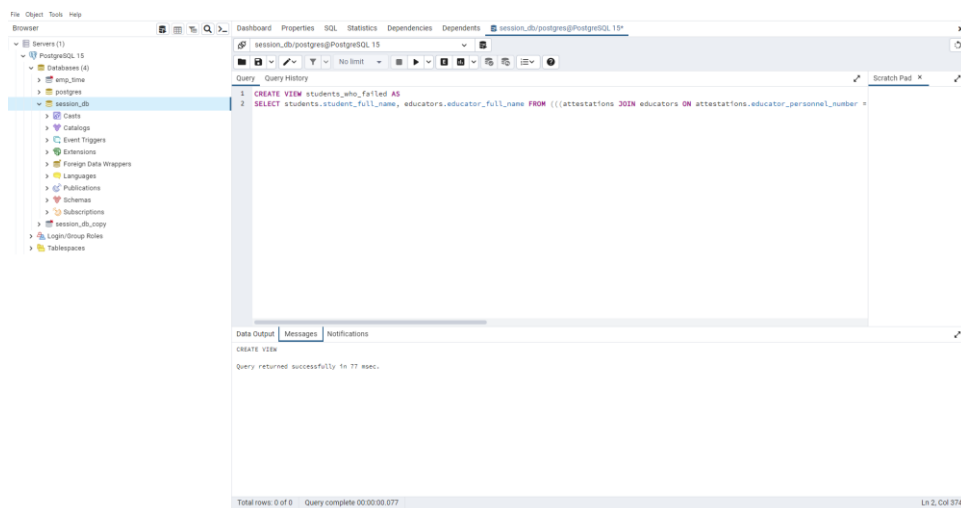
1

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

Команда:

```
CREATE VIEW students_who_failed AS  
SELECT students.student_full_name,  
        educators.educator_full_name  
FROM (((attestations  
        JOIN educators ON  
attestations.educator_personnel_number =  
educators.educator_personnel_number)  
        JOIN st_in_group ON  
attestations.student_personnel_number =  
st_in_group.student_personnel_number)  
        JOIN students ON st_in_group.student_id =  
students.student_id)  
WHERE attestations.mark = '2F';
```

Результат:



Содержимое:

	student_full_name character varying (100)	educator_full_name character varying (100)
1	Киселева Варвара Павловна	Нестеров Сергей Сергеевич
2	Богданова Анастасия Константиновна	Нестеров Сергей Сергеевич
3	Богданова Анастасия Константиновна	Гусев Вадим Андреевич
4	Богданова Анастасия Константиновна	Иванова Алиса Матвеевна

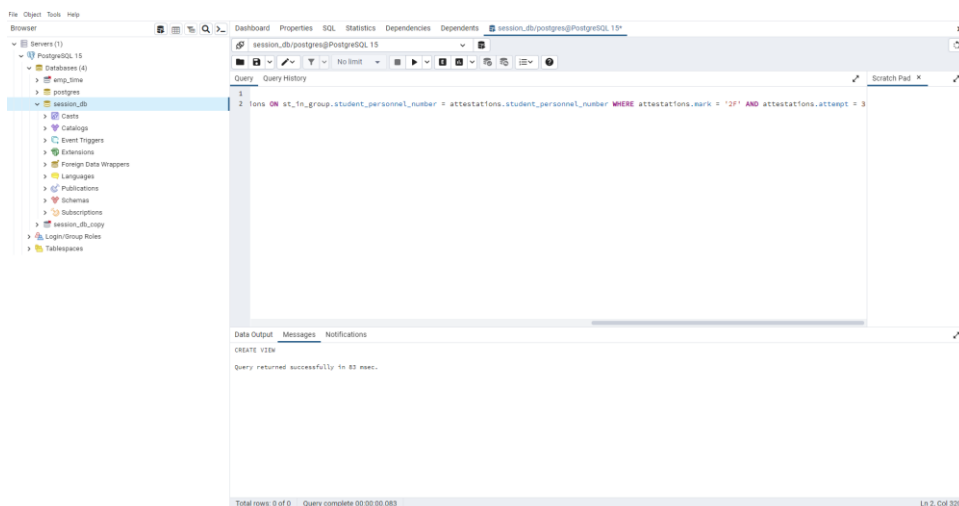
2

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

Команда:

```
CREATE VIEW students_who_failed_really_hard AS
SELECT DISTINCT students.student_full_name,
                st_in_group.student_personnel_number
FROM (st_in_group
      JOIN students ON st_in_group.student_id =
students.student_id)
JOIN attestations ON st_in_group.student_personnel_number =
attestations.student_personnel_number
WHERE attestations.mark = '2F'
      AND attestations.attempt = 3
```

Результат:



Содержимое:

	student_full_name character varying (100) 🔒	student_personnel_number integer 🔒
1	Богданова Анастасия Константиновна	333004

Запросы на модификацию данных

1

Формулировка: Вставить лишние данные об аттестации, зная имя студента и имя преподавателя в таблицу attestations.

Команда:

```
INSERT INTO attestations (attestation_id,  
student_personnel_number, educator_personnel_number,  
disc_in_curr_id, mark, attempt)  
  
VALUES (25, (SELECT st_in_group.student_personnel_number FROM  
st_in_group JOIN students ON st_in_group.student_id =  
students.student_id WHERE students.student_full_name = 'Агеева  
Вероника Владиславовна'), (SELECT educator_personnel_number  
FROM educators WHERE educator_full_name = 'Муфалали  
Чипо Мубалу'), 4, 'Незачёт', 1)
```

Было:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
15	333004	111333	3	2F	15	1
16	333004	111555	4	Зачёт	16	1
17	333005	111111	2	5A	17	1
18	333005	111222	1	5A	18	1
19	333005	111333	3	5A	19	1
20	333005	111444	4	Зачёт	20	1
21	333006	111111	2	4C	21	1
22	333006	111222	1	4B	22	1
23	333006	111333	3	5A	23	1
24	333006	111444	4	Зачёт	24	1
25	333004	111222	1	3D	25	2
26	333004	111111	2	2F	26	2
27	333004	111333	3	3D	27	2
28	333004	111111	2	2F	28	3

Стало:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
16	333004	111555	4	Зачёт	16	1
17	333005	111111	2	5A	17	1
18	333005	111222	1	5A	18	1
19	333005	111333	3	5A	19	1
20	333005	111444	4	Зачёт	20	1
21	333006	111111	2	4C	21	1
22	333006	111222	1	4B	22	1
23	333006	111333	3	5A	23	1
24	333006	111444	4	Зачёт	24	1
25	333004	111222	1	3D	25	2
26	333004	111111	2	2F	26	2
27	333004	111333	3	3D	27	2
28	333004	111111	2	2F	28	3
29	333005	111555	4	Незачёт	29	1

2

Формулировка: Изменить номер попытку конкретного человека, сдававшего математику в таблице attestations.

Команда:

UPDATE attestations

SET attempt = 2

WHERE student_personnel_number = 333004

AND disc_in_curr_id =

SELECT disc_in_curr_id

FROM disc_in_curr

JOIN disciplines **ON** disc_in_curr.discipline_id =

disciplines.discipline_id **WHERE** discipline_name = 'Математика'

Было:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
8	333002	111444	4	Зачёт	8	1
9	333003	111111	2	5A	9	1
10	333003	111222	1	5A	10	1
11	333003	111333	3	5A	11	1
12	333003	111444	4	Зачёт	12	1
13	333004	111111	2	2F	13	1
14	333004	111222	1	2F	14	1
15	333004	111333	3	2F	15	1
16	333004	111555	4	Зачёт	16	1
17	333005	111111	2	5A	17	1
18	333005	111222	1	5A	18	1
19	333005	111333	3	5A	19	1
20	333005	111444	4	Зачёт	20	1
21	333006	111111	2	4C	21	1

Total rows: 20 of 20 Query complete 00:00:00.128

Стало:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
1	333001	111111	2	5A	1	1
2	333001	111222	1	4B	2	1
3	333001	111333	3	5A	3	1
4	333001	111444	4	Зачёт	4	1
5	333002	111111	2	2F	5	1
6	333002	111222	1	4B	6	1
7	333002	111333	3	4C	7	1
8	333002	111444	4	Зачёт	8	1
9	333003	111111	2	5A	9	1
10	333003	111222	1	5A	10	1
11	333003	111333	3	5A	11	1
12	333003	111444	4	Зачёт	12	1
13	333004	111111	2	2F	13	1
14	333004	111222	1	2F	14	1

Total rows: 20 of 20 Query complete 00:00:00.185

3

Формулировка: Удалить запись из пункта 1, используя имя студента и имя преподавателя.

Команда:

```
DELETE
FROM attestations
WHERE student_personnel_number =
      (SELECT st_in_group.student_personnel_number
       FROM st_in_group
       JOIN students ON st_in_group.student_id =
students.student_id
```

```

WHERE students.student_full_name = 'Агеева Вероника
Владиславовна')
AND educator_personnel_number =
(SELECT educator_personnel_number
FROM educators
WHERE educator_full_name = 'Муфалали Чипо Мубалу')

```

Было:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
16	333004	111555	4	Зачёт	16	1
17	333005	111111	2	5A	17	1
18	333005	111222	1	5A	18	1
19	333005	111333	3	5A	19	1
20	333005	111444	4	Зачёт	20	1
21	333006	111111	2	4C	21	1
22	333006	111222	1	4B	22	1
23	333006	111333	3	5A	23	1
24	333006	111444	4	Зачёт	24	1
25	333004	111222	1	3D	25	2
26	333004	111111	2	2F	26	2
27	333004	111333	3	3D	27	2
28	333004	111111	2	2F	28	3
29	333005	111555	4	Незачёт	29	1

Стало:

	student_personnel_number integer	educator_personnel_number integer	disc_in_curr_id integer	mark character varying (10)	attestation_id [PK] integer	attempt integer
15	333004	111333	3	2F	15	1
16	333004	111555	4	Зачёт	16	1
17	333005	111111	2	5A	17	1
18	333005	111222	1	5A	18	1
19	333005	111333	3	5A	19	1
20	333005	111444	4	Зачёт	20	1
21	333006	111111	2	4C	21	1
22	333006	111222	1	4B	22	1
23	333006	111333	3	5A	23	1
24	333006	111444	4	Зачёт	24	1
25	333004	111222	1	3D	25	2
26	333004	111111	2	2F	26	2
27	333004	111333	3	3D	27	2
28	333004	111111	2	2F	28	3

Индексы

1

Мы попытаемся выполнить следующий запрос без индекса:

```
SELECT *  
FROM students  
WHERE student_full_name = 'AAA AAA AAA'
```

План выполнения запроса:



Результат работы:

22.05.2023 13:32:08	6	51 msec
Date	Rows affected	Duration

Copy Copy to Query Editor

```
SELECT * FROM students WHERE student_full_name = 'AAA AAA AAA'
```

Messages

Successfully run. Total query runtime: 51 msec. 6 rows affected.

Теперь создадим индекс под этот запрос - индекс по столбцу `student_full_name`:

```
CREATE INDEX idx_student_full_name ON students  
(student_full_name)
```

Новый план выполнения того же запроса:



Новый результат выполнения того же запроса:


22.05.2023 13:36:43	6	106 msec
Date	Rows affected	Duration
<div>Copy Copy to Query Editor</div> <pre>SELECT * FROM students WHERE student_full_name = 'AAA AAA AAA'</pre>		
Messages Successfully run. Total query runtime: 106 msec. 6 rows affected.		

2

Мы попытаемся выполнить следующий запрос без индекса:

```
SELECT *
FROM auditoriums
WHERE auditorium_capacity > 50
AND auditorium_type = 'Лекционный зал'
```

План выполнения запроса:

 auditoriums	<div>✓ Successfully run. Total query runtime: 88 msec. 1 row affected. ✓</div>
--	--

Результат работы:

22.05.2023 13:49:16	16562	83 msec
Date	Rows affected	Duration
<div>Copy Copy to Query Editor</div> <pre>SELECT * FROM auditoriums WHERE auditorium_capacity > 50 AND auditorium_type = 'Лек'</pre>		
Messages Successfully run. Total query runtime: 83 msec. 16562 rows affected.		

Теперь создадим составной индекс по столбцам auditorium_capacity и auditorium_type:

```
CREATE INDEX idx_capacity_and_type ON auditoriums
(auditorium_capacity, auditorium_type)
```

Новый план выполнения того же запроса:



Новый результат выполнения того же запроса:

22.05.2023 13:53:00	16562	69 msec
Date	Rows affected	Duration

Copy

Copy to Query Editor

```
SELECT * FROM auditoriums WHERE auditorium_capacity > 50 AND auditorium_type = 'Лект
```

Messages

Successfully run. Total query runtime: 69 msec. 16562 rows affected.

Выводы

Я овладел практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.