Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами» по дисциплине «Проектирование и реализация баз данных» Вариант №2 - БД «Сессия»

Автор: Клименков В.М.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

1. Запросы к базе данных	. 3
2. Представления	. 8
3. Запросы на модификацию данных	. 9
4. Query History и графическое представление запросов	. 11
5. Индексы	12
Вывод	12

Цель работы

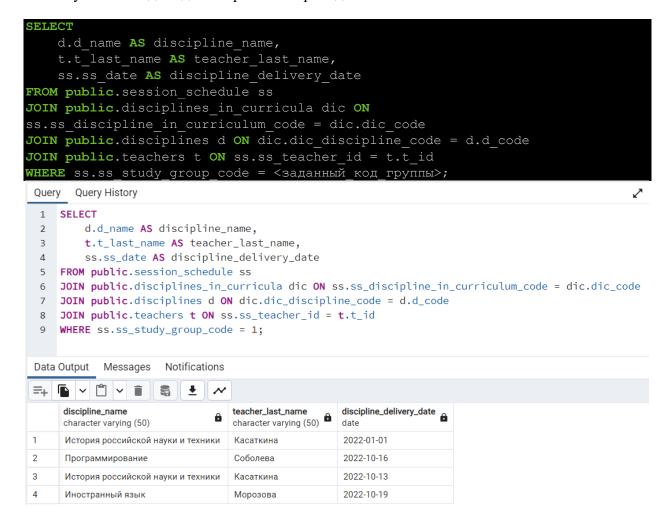
Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

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.

1. Запросы к базе данных

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



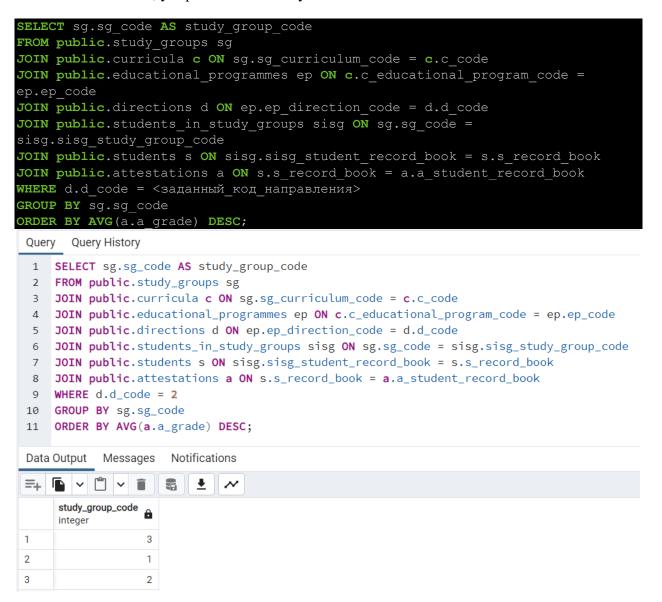
• Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```
SELECT
    a.a student record book AS student record book,
    s.s first name AS student_first_name,
    s.s last name AS student last name,
    t.t last name AS teacher last name
FROM public.attestations a
JOIN public.students s ON a.a student record book = s.s record book
JOIN public.teachers t ON a.a_teacher_id = t.t_id
WHERE a.a attempt = 1 AND a.a grade = '2';
        Query History
 Query
      SELECT
  1
          a.a_student_record_book AS student_record_book,
  2
          s.s_first_name AS student_first_name,
  3
          s.s_last_name AS student_last_name,
  4
  5
          t.t_last_name AS teacher_last_name
      FROM public.attestations a
  6
      JOIN public.students s ON a.a_student_record_book = s.s_record_book
  7
      JOIN public.teachers t ON a.a_teacher_id = t.t_id
  8
      WHERE a.a_attempt = 1 AND a.a_grade = '2';
 Data Output
                         Notifications
              Messages
 =+
                                             student_last_name
      student_record_book
                          student_first_name
                                                                 teacher_last_name
                                                                 character varying (50)
      integer
                          character varying (50)
                                             character varying (50)
 1
                  381030
                          Арина
                                              Круглова
                                                                 Николаев
 2
                                              Егорова
                  516111
                          Моника
                                                                  Соболева
```

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



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



• 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.

```
SELECT
     s.s record book AS student record book,
     s.s first name AS student first name,
     s.s last name AS student last name,
     sisg.sisg study group code AS study group code,
     AVG(a.a grade) AS average grade,
     CASE WHEN MIN(a.a grade) > 3 THEN 1 ELSE 0 END AS receives scholarship
FROM public.students s
JOIN public.students in study groups sisg ON s.s record book =
sisg.sisg student record book
JOIN public.attestations a ON s.s record book = a.a student record book
GROUP BY
     s.s record book,
     s.s first name,
     s.s last name,
     sisg.sisg study group code
ORDER BY
     sisg.sisg study group code,
     s.s last name;
 Query Query History
 1 SELECT
        s.s_record_book AS student_record_book,
 3
        s.s_first_name AS student_first_name,
 4
        s.s_last_name AS student_last_name,
 5
        sisg.sisg_study_group_code AS study_group_code,
        AVG(a.a_grade) AS average_grade,
 6
 7
        CASE WHEN MIN(a.a_grade) > 3 THEN 1 ELSE 0 END AS receives_scholarship
 8
    FROM public.students s
 9
    JOIN public.students_in_study_groups sisg ON s.s_record_book = sisg.sisg_student_record_book
 JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
 11 GROUP BY
        s.s_record_book,
 12
 13
        s.s_first_name,
        s.s_last_name.
 14
 15
        sisg.sisg_study_group_code
 16 ORDER BY
 17
        sisg.sisg_study_group_code,
 18
        s.s_last_name;
 Data Output Messages Notifications
=+ 🖺 🗸 📋 🗸 📋
                   S + ~
     student_record_book
                                                                                    receives_scholarship
                     student_first_name
                                      student_last_name
                                                      study_group_code
                                                                     average_grade
                                                                  â
                     character varying (50)
                                                   a
                                                                                 a
                                      character varying (50)
     integer
                                                      integer
                                                                    numeric
                                                                                    integer
               516111
                                                                     2.50000000000000000
                                                                                                 0
2
               742645 Лука
                                                                  1
                                                                     4.00000000000000000
                                                                                                  1
                                      Семенов
3
               400419
                     Иван
                                      Чернышев
                                                                  1
                                                                     4.50000000000000000
                                                                                                  1
               252108 Григорий
                                      Косарев
                                                                     5.00000000000000000
                                                                                                  1
               381030 Арина
                                                                                                 0
                                                                     3.50000000000000000
                                      Круглова
```

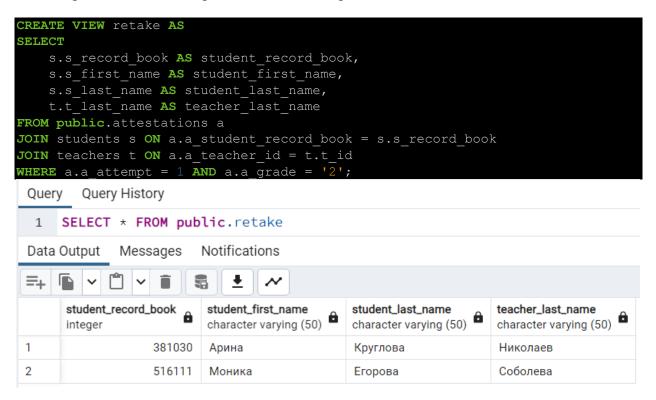
• Вывести список студентов, получивших максимальный средний балл в своей группе.

```
SELECT
       s.s record book AS student record book,
       s.s first name AS student first name,
       s.s last name AS student last name
FROM public.students s
JOIN public.students in study groups sisg ON s.s record book =
sisg.sisg student record book
JOIN (
       SELECT
              a.a student record book AS r student record book,
              RANK() OVER (PARTITION BY sisg.sisg study group code ORDER BY
AVG(a.a grade) DESC) AS rank
       FROM public.attestations a
       JOIN public.students_in_study groups sisg ON a.a student record book =
sisg.sisg student record book
       GROUP BY a.a student record book, sisg.sisg study group code
  AS r ON s.s record book = r.r student record book
WHERE r.rank = 1
 Query Query History
  1 SELECT
           s.s_record_book AS student_record_book,
           s.s_first_name AS student_first_name,
  3
  4
           s.s_last_name AS student_last_name
  5 FROM public.students s
  6 JOIN public.students_in_study_groups sisg ON s.s_record_book = sisg.sisg_student_record_book
  7 JOIN (
  8
           SELECT
  9
                 a.a_student_record_book AS r_student_record_book,
 10
                 RANK() OVER (PARTITION BY sisg.sisg_study_group_code ORDER BY AVG(a.a_grade) DESC) AS rank
 11
           FROM public.attestations a
 12
            JOIN public.students_in_study_groups sisg ON a.a_student_record_book = sisg.sisg_student_record_book
 13
           GROUP BY a.a_student_record_book, sisg.sisg_study_group_code
 14 ) AS r ON s.s_record_book = r.r_student_record_book
 15 WHERE r.rank = 1
 Data Output Messages Notifications

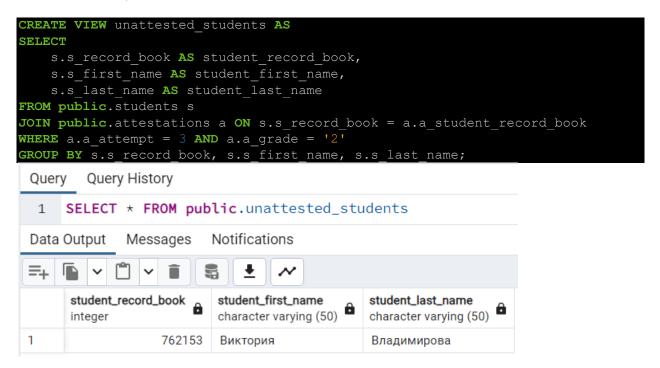
    →
    □
    ✓
    □
    □
    ✓
    □
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓</
       student_record_book integer student_first_name character varying (50) character varying (50) character varying (50)
                     252108 Григорий
                                                      Косарев
2
                     400419 Иван
                                                      Чернышев
3
                     534421 Пётр
                                                      Смирнов
```

2. Представления

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

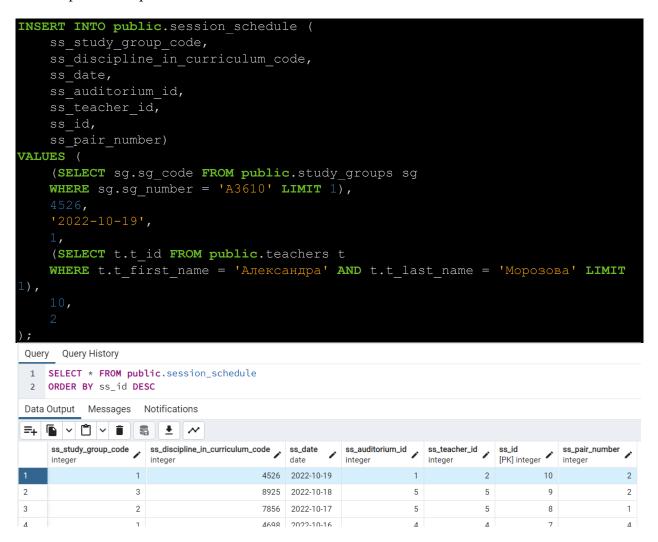


• Данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

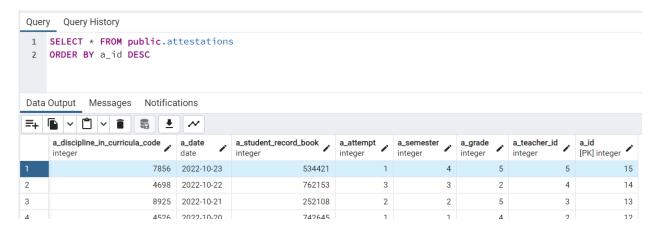


3. Запросы на модификацию данных

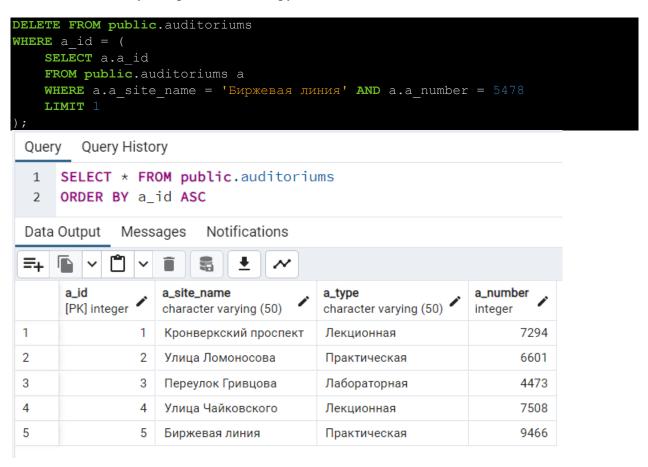
• Вставка нового экзамена в расписание сессии по номеру аудитории и имени и фамилии преподавателя.



• Изменение оценки студента по его имени и фамилии, а также названию предмета.

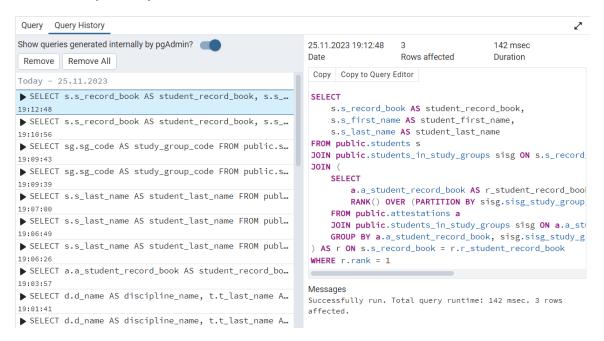


• Удаление аудитории по её номеру и площадке.

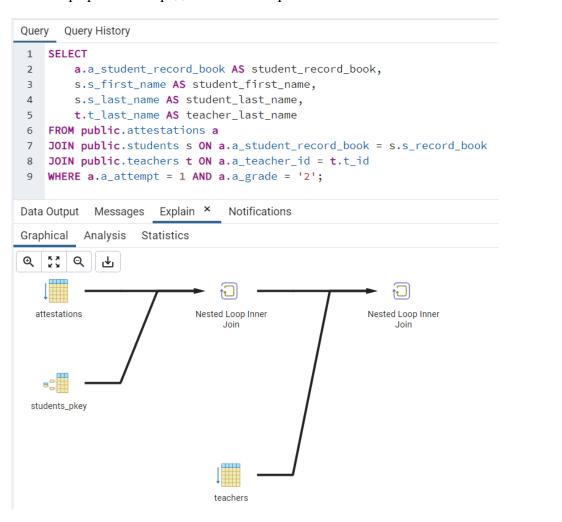


4. Query History и графическое представление запросов

Query History



• Графическое представление запроса



5. Индексы

```
CREATE INDEX idx_disciplines
ON public.disciplines (d_name);
CREATE INDEX idx_teachers
ON public.teachers (t_last_name);
CREATE INDEX idx_session_schedule
ON public.session_schedule (ss_date, ss_study_group_code);
```

• Без индексов:

```
Query Query History
1 SELECT
       d.d_name AS discipline_name,
2
3
        t.t_last_name AS teacher_last_name,
       ss.ss_date AS discipline_delivery_date
4
5
   FROM public.session_schedule ss
   JOIN public.disciplines_in_curricula dic ON ss.ss_discipline_in_curriculum_code = dic.dic_code
   JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
   JOIN public.teachers t ON ss.ss_teacher_id = t.t_id
9 WHERE ss.ss_study_group_code = 1;
Data Output Messages Notifications
Successfully run. Total query runtime: 160 msec.
4 rows affected.
```

• С индексами:

```
Query Query History
1 SELECT
2
        d.d_name AS discipline_name,
3
        t.t_last_name AS teacher_last_name,
4
        ss.ss_date AS discipline_delivery_date
5 FROM public.session_schedule ss
6 JOIN public.disciplines_in_curricula dic ON ss.ss_discipline_in_curriculum_code = dic.dic_code
7 JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
8 JOIN public.teachers t ON ss.ss teacher id = t.t id
9 WHERE ss.ss_study_group_code = 1;
Data Output Messages Notifications
Successfully run. Total query runtime: 72 msec.
4 rows affected.
```

Вывод

В данной лабораторной работе мне удалось создать требуемые запросы и представления к разработанной раннее базе данных, составить три запроса на модификацию данных (INSERT, UPDATE, DELETE) с использование подзапросов, изучить графическое представление запросов и просмотреть историю запросов, а также создать индексы для запроса и сравнить время выполнения запроса без индексов и с индексами.