

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**»

Автор: Никифорова Анна Дмитриевна

Факультет: Инфокоммуникационные технологии

Группа: K32421

Преподаватель: Говорова Марина Михайловна



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2023

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.

Индивидуальное практическое задание:

Предметная область «Школа».

Составьте запросы на выборку:

1. Найти все кабинеты без проектора, кол-во мест в которых ≥ 30 , которые максимально заняты по текущему расписанию
2. Найти всех преподавателей, которые имеют больше одного образования
3. Создать таблицу с ФИО препода и предметами, которые он преподаёт, отсортировать по нагрузке по этим предметам
4. Найти учеников с максимальным средним баллом по школе
5. Расчет среднего балла каждого ученика по каждому предмету
6. Найти в каждом классе учеников с максимальным и минимальным средним баллом, указать отдельным полем название статистики (минимум/максимум)
7. Найти день с максимальным кол-вом уроков для 10А в текущем учебном году (предполагается, что расписание составлено на год)
8. Составить список учащихся, которые не имеют ни одной оценки по информатике
9. Составить таблицу, содержащую имя класса в формате буква_цифра, ФИО классного руководителя и ФИО старосты

Создайте представления:

1. Количество человек в классе (добавить поле в таблицку Класс)
2. Расчет общей нагрузки преподавателя (добавить поле в таблицку Преподаватель)

Схема базы данных:

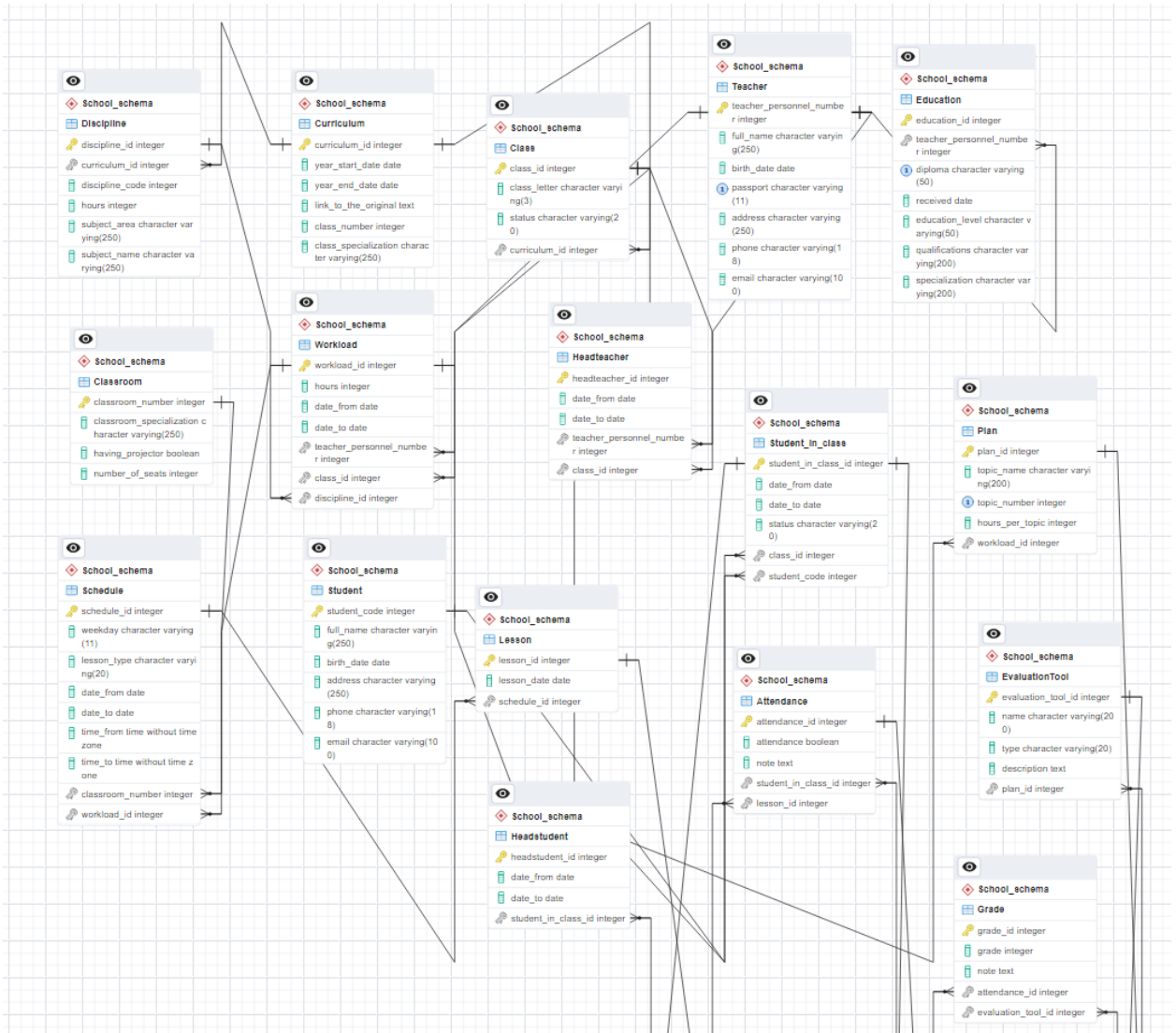


Рис. 1 – Схема базы данных

Выполнение задания:

Все запросы по умолчанию выполняются в рамках одного учебного года, если в задании не указано иное.

SELECT

1. Найти все кабинеты без проектора, кол-во мест в которых ≥ 30 , которые максимально заняты по текущему расписанию

SELECT *

```

FROM "School_schema"."Classroom"
WHERE "School_schema"."Classroom".classroom_number IN
  (SELECT classroom_number
   FROM
     (SELECT classroom_number,
              COUNT(*)
       FROM "School_schema"."Schedule"
       JOIN "School_schema"."Classroom" USING(classroom_number)
       WHERE date_from <= now()
             AND date_to >= now()
             AND number_of_seats >= 30
             AND NOT having_projector
       GROUP BY classroom_number) AS TEMP1
   WHERE TEMP1.count =
     (SELECT MAX(TEMP1.count)
      FROM
        (SELECT COUNT(*)
         FROM "School_schema"."Schedule"
         JOIN "School_schema"."Classroom" USING(classroom_number)
         WHERE date_from <= now()
               AND date_to >= now()
               AND number_of_seats >= 30
               AND NOT having_projector
         GROUP BY classroom_number) AS TEMP1));

```

Data Output

Messages

Notifications

≡

📄

▼

📋

🗑️

🗄️

⬇️

📈

	<div>classroom_number</div> <div>[PK] integer</div>	<div>classroom_specialization</div> <div>character varying (250)</div>	<div>having_projector</div> <div>boolean</div>	<div>number_of_seats</div> <div>integer</div>
1	39	[null]	false	31

Total rows: 1 of 1

Query complete 00:00:00.057

Рис. 2 – SELECT 1

2. Найти всех преподавателей, которые имеют больше одного образования

```

SELECT *
FROM
  (SELECT teacher_personnel_number
   FROM "School_schema"."Education"
   GROUP BY teacher_personnel_number
   HAVING COUNT(*) > 1) AS TEMP1
JOIN "School_schema"."Teacher" USING(teacher_personnel_number);

```

Data Output Messages Notifications								
	teacher_personnel_number [PK] integer	full_name character varying (250)	birth_date date	passport character varying (11)	address character varying (250)	phone character varying (18)	email character varying (100)	
1	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 3...	+7 (919) 492-06-48	canadian1815@example.com	
2	11	Зотьева Лера	1992-12-14	7723 217197	[null]	+7 (296) 531-57-36	circumstances1976@outlook.com	
3	25	Шамбарова Эсмירה	1982-05-11	3885 520883	[null]	+7 (636) 439-07-12	[null]	
Total rows: 3 of 3 Query complete 00:00:00.060 Ln 1,6								

Рис. 3 – SELECT 2

3. Создать таблицу с ФИО препода и предметами, которые он преподает, отсортировать по нагрузке по этим предметам

```

SELECT full_name,
       subject_name,
       SUM(hours) AS hours
FROM (
    (SELECT *
     FROM "School_schema"."Teacher"
     JOIN "School_schema"."Workload" USING (teacher_personnel_number)
     WHERE '01.09.2022' >= date_from
        AND date_from <= '31.08.2023') AS TEMP1
    JOIN
    (SELECT discipline_id,
            subject_name
     FROM "School_schema"."Discipline") AS TEMP2 USING (discipline_id))
GROUP BY teacher_personnel_number,
         full_name,
         discipline_id,
         subject_name
ORDER BY hours;
```

Data Output Messages Notifications				
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>				
	full_name character varying (250)	subject_name character varying (250)	hours bigint	
1	Волкова Жасмин	Физическая культура	36	
2	Ланькова Теона	История России	36	
3	Нестерова Арабелла	История России	36	
4	Котляревский Ефим	Литература	36	
5	Александров Тарас	География	36	
6	Нестерова Арабелла	Литература	36	
7	Норова Фелисити	Биология	36	
8	Катерли Серафим	Обществознание	36	
9	Распутина Майя	Физическая культура	36	
10	Банщикова Чеслава	Математика	36	
11	Кузнецова Ольга	Физическая культура	36	
12	Квартальнова Томирис	Английский язык	36	
13	Обросимов Мина	Русский язык	36	
14	Марковчин Кирилл	Геометрия	36	
15	Шамбарова Эмира	Биология	36	
16	Распутина Майя	Математика	36	
17	Квартальнова Томирис	Алгебра	36	
Total rows: 1000 of 1585		Query complete 00:00:00.073		

Рис. 4 – SELECT 3, верхняя часть таблицы

Data Output Messages Notifications			
	full_name character varying (250)	subject_name character varying (250)	hours bigint
1569	Крупина Айсель	Литература	72
1570	Басманова Ильзира	Русский язык	72
1571	Котляревский Ефим	Английский язык	72
1572	Евлампиев Всеволод	Физическая культура	72
1573	Клепов Геннадий	История России	72
1574	Александров Тарас	Химия	72
1575	Крупина Айсель	Английский язык	72
1576	Кириллова Милада	Математика	72
1577	Карпенков Аристарх	Геометрия	72
1578	Янин Никодим	Физическая культура	72
1579	Володарская Раиля	Русский язык	72
1580	Зотьева Лера	Русский язык	108
1581	Катерли Серафим	Математика	108
1582	Распутина Майя	Математика	108
1583	Колядин Варсонофий	Русский язык	108
1584	Кириллова Милада	Физическая культура	108
1585	Котляревский Ефим	Английский язык	108
Total rows: 1585 of 1585		Query complete 00:00:00.073	

Рис. 5 – SELECT 3, нижняя часть таблицы

4. Найти учеников с максимальным средним баллом по школе

```

SELECT *
FROM
  (SELECT student_in_class_id,
    AVG(grade) AS mean
  FROM
    (SELECT *
      FROM "School_schema"."Curriculum"
      JOIN "School_schema"."Class" USING(curriculum_id)
      JOIN "School_schema"."Student_in_class" USING(class_id)
      WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY student_in_class_id) AS TEMP2
WHERE mean =
  (SELECT MAX(mean)
    FROM
      (SELECT student_in_class_id,

```



```
JOIN "School_schema"."Grade" USING(attendance_id,
                                   evaluation_tool_id)

GROUP BY student_in_class_id,
         discipline_id,
         subject_name;
```

Data Output	Messages	Graph Visualiser	×	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>				
	student_in_class_id integer	subject_name character varying (250)	mean numeric	
1	2458	Русский язык	4.1375000000000000	
2	2458	Литература	4.0232558139534884	
3	2458	Английский язык	4.2500000000000000	
4	2458	Математика	4.0657894736842105	
5	2458	История России	4.2000000000000000	
6	2458	География	4.1578947368421053	
7	2458	Биология	4.2000000000000000	
8	2458	Физическая культура	3.9666666666666667	
9	2505	Русский язык	4.0149253731343284	
10	2505	Литература	4.0000000000000000	
11	2505	Английский язык	4.0540540540540541	
12	2505	Математика	3.9302325581395349	
13	2505	История России	3.9642857142857143	
14	2505	География	4.3076923076923077	
15	2505	Биология	3.9130434782608696	
16	2505	Физическая культура	4.0238095238095238	
17	2527	Русский язык	3.9402985074626866	
18	2527	Литература	3.8947368421052632	
19	2527	Английский язык	4.0500000000000000	
Total rows: 185 of 185		Query complete 00:02:37.514		

Рис. 7 – SELECT 5

6. Найти в каждом классе учеников с максимальным и минимальным средним баллом, указать отдельным полем название статистики (минимум/максимум)

```
SELECT class_id,
       mn_student_in_class_id AS student_in_class_id,
       ROUND(minimum, 2) AS val,
       'min' AS stats
FROM
  (SELECT class_id,
         student_in_class_id AS mn_student_in_class_id,
         minimum
   FROM
     (SELECT class_id,
            student_in_class_id,
```

```

        AVG(grade) AS mean
FROM
    (SELECT *
      FROM "School_schema"."Curriculum"
     JOIN "School_schema"."Class" USING(curriculum_id)
     JOIN "School_schema"."Student_in_class" USING(class_id)
     WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY class_id,
              student_in_class_id) AS TABLE1
JOIN
    (SELECT class_id,
           MIN(mean) AS minimum
      FROM
        (SELECT class_id,
                 student_in_class_id,
                 AVG(grade) AS mean
          FROM
            (SELECT *
              FROM "School_schema"."Curriculum"
             JOIN "School_schema"."Class" USING(curriculum_id)
             JOIN "School_schema"."Student_in_class" USING(class_id)
             WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS T
EMP1
            JOIN "School_schema"."Attendance" USING(student_in_class_id)
            JOIN "School_schema"."Grade" USING(attendance_id)
            GROUP BY class_id,
                     student_in_class_id) AS AVG_TABLE
        GROUP BY class_id) AS TABLE2 USING(class_id)
    WHERE mean = minimum) AS AAA1
JOIN
    (SELECT class_id,
           student_in_class_id AS mx_student_in_class_id,
           maximum
      FROM
        (SELECT class_id,
                 student_in_class_id,
                 AVG(grade) AS mean
          FROM
            (SELECT *
              FROM "School_schema"."Curriculum"
             JOIN "School_schema"."Class" USING(curriculum_id)
             JOIN "School_schema"."Student_in_class" USING(class_id)
             WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
1
            JOIN "School_schema"."Attendance" USING(student_in_class_id)

```

```

        JOIN "School_schema"."Grade" USING(attendance_id)
        GROUP BY class_id,
                 student_in_class_id) AS TABLE1
JOIN
  (SELECT class_id,
         MAX(mean) AS maximum
   FROM
     (SELECT class_id,
            student_in_class_id,
            AVG(grade) AS mean
      FROM
        (SELECT *
         FROM "School_schema"."Curriculum"
         JOIN "School_schema"."Class" USING(curriculum_id)
         JOIN "School_schema"."Student_in_class" USING(class_id)
         WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS T
      WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS T
    EMP1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY class_id,
             student_in_class_id) AS AVG_TABLE
  GROUP BY class_id) AS TABLE2 USING(class_id)
  WHERE mean = maximum) AS AAA2 USING(class_id)
UNION
SELECT class_id,
       mx_student_in_class_id AS student_in_class_id,
       ROUND(maximum, 2) AS val,
       'max' AS stats
FROM
  (SELECT class_id,
         student_in_class_id AS mn_student_in_class_id,
         minimum
   FROM
     (SELECT class_id,
            student_in_class_id,
            AVG(grade) AS mean
      FROM
        (SELECT *
         FROM "School_schema"."Curriculum"
         JOIN "School_schema"."Class" USING(curriculum_id)
         JOIN "School_schema"."Student_in_class" USING(class_id)
         WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
      WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
    1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY class_id,
             student_in_class_id) AS TABLE1
JOIN

```

```

(SELECT class_id,
        MIN(mean) AS minimum
FROM
    (SELECT class_id,
            student_in_class_id,
            AVG(grade) AS mean
    FROM
        (SELECT *
        FROM "School_schema"."Curriculum"
        JOIN "School_schema"."Class" USING(curriculum_id)
        JOIN "School_schema"."Student_in_class" USING(class_id)
        WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS T
EMP1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY class_id,
            student_in_class_id) AS AVG_TABLE
GROUP BY class_id) AS TABLE2 USING(class_id)
WHERE mean = minimum) AS AAA1
JOIN
    (SELECT class_id,
            student_in_class_id AS mx_student_in_class_id,
            maximum
    FROM
        (SELECT class_id,
            student_in_class_id,
            AVG(grade) AS mean
    FROM
        (SELECT *
        FROM "School_schema"."Curriculum"
        JOIN "School_schema"."Class" USING(curriculum_id)
        JOIN "School_schema"."Student_in_class" USING(class_id)
        WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
1
    JOIN "School_schema"."Attendance" USING(student_in_class_id)
    JOIN "School_schema"."Grade" USING(attendance_id)
    GROUP BY class_id,
            student_in_class_id) AS TABLE1
JOIN
    (SELECT class_id,
            MAX(mean) AS maximum
    FROM
        (SELECT class_id,
            student_in_class_id,
            AVG(grade) AS mean
    FROM
        (SELECT *
        FROM "School_schema"."Curriculum"

```

```

        JOIN "School_schema"."Class" USING(curriculum_id)
        JOIN "School_schema"."Student_in_class" USING(class_id)
        WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS T
EMP1
        JOIN "School_schema"."Attendance" USING(student_in_class_id)
        JOIN "School_schema"."Grade" USING(attendance_id)
        GROUP BY class_id,
                 student_in_class_id) AS AVG_TABLE
        GROUP BY class_id) AS TABLE2 USING(class_id)
        WHERE mean = maximum) AS AAA2 USING(class_id)
ORDER BY class_id;

```

	class_id integer	student_in_class_id integer	val numeric	stats text
1	77	2527	4.02	min
2	77	2458	4.11	max
3	79	2505	4.00	max
4	79	2541	3.98	min
5	81	2617	3.98	max
6	81	2617	3.98	min
7	82	2594	3.98	min
8	82	2594	3.98	max
9	84	2706	3.88	min
10	84	2713	4.00	max
11	85	2704	4.02	max
12	85	2649	3.98	min
13	86	2797	3.98	max
14	86	2797	3.98	min
15	87	2745	3.88	min
16	87	2791	4.03	max
17	88	2747	4.01	min
18	88	2747	4.01	max
19	89	2896	3.99	min
20	89	2896	3.99	max
21	90	2904	4.04	max
22	90	2904	4.04	min
23	92	2954	4.00	min
Total rows: 26 of 26			Query complete 00:00:01.396	

Рис. 8 – SELECT 6

7. Найти день с максимальным кол-вом уроков для 10А в текущем учебном году (предполагается, что расписание составлено на год)

```

SELECT weekday,
       how_many_lessons
FROM
  (SELECT weekday,
           COUNT(*) AS how_many_lessons

```

```

FROM
  (SELECT *
   FROM
     (SELECT *
      FROM "School_schema"."Curriculum"
      WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023'
      AND class_number = 10) AS TEMP1
     JOIN "School_schema"."Class" USING(class_id)
     WHERE class_letter = 'A') AS TEMP2
   JOIN "School_schema"."Workload" USING(class_id)
   JOIN "School_schema"."Schedule" USING(workload_id)
   GROUP BY weekday) AS AAA1
WHERE how_many_lessons =
  (SELECT MAX(how_many_lessons)
   FROM
     (SELECT weekday,
      COUNT(*) AS how_many_lessons
     FROM
       (SELECT *
        FROM
          (SELECT *
           FROM "School_schema"."Curriculum"
           WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023'
           AND class_number = 10) AS TEMP1
          JOIN "School_schema"."Class" USING(class_id)
          WHERE class_letter = 'A') AS TEMP2
          JOIN "School_schema"."Workload" USING(class_id)
          JOIN "School_schema"."Schedule" USING(workload_id)
          GROUP BY weekday) AS AAA2);

```

Data Output

Messages

Notifications

	<div>weekday</div> <div>character varying (11)</div>	<div>how_many_lessons</div> <div>bigint</div>
1	Среда	5

Рис. 9 – SELECT 7

8. Составить список учащихся, которые не имеют ни одной оценки по информатике

```

SELECT student_in_class_id
FROM (
  (SELECT *
   FROM "School_schema"."Curriculum"

```

```

WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023') AS TEMP
1
JOIN "School_schema"."Class" USING(curriculum_id)
JOIN "School_schema"."Student_in_class" USING(class_id)) AS AAA1
WHERE NOT EXISTS
(SELECT *
FROM (
(SELECT *
FROM
(SELECT *
FROM "School_schema"."Curriculum"
WHERE year_start_date BETWEEN '01.09.2022' AND '31.08.2023')
AS TEMP1
JOIN "School_schema"."Class" USING(curriculum_id)
JOIN "School_schema"."Student_in_class" USING(class_id)
JOIN "School_schema"."Discipline" USING(curriculum_id)
WHERE subject_name = 'Информатика') AS TEMP2
JOIN "School_schema"."Workload" USING(class_id,
discipline_id)
JOIN "School_schema"."Schedule" USING(workload_id)
JOIN "School_schema"."Lesson" USING(schedule_id)
JOIN "School_schema"."Plan" USING(workload_id)
JOIN "School_schema"."EvaluationTool" USING(plan_id)
JOIN "School_schema"."Attendance" USING(lesson_id,
student_in_class_id)
JOIN "School_schema"."Grade" USING(attendance_id,
evaluation_tool_id)) AS AAA2
WHERE AAA1.student_in_class_id = AAA2.student_in_class_id);

```


Data Output		Messages	Notifications
	student_in_class_id [PK] integer		
1	2443		
2	2444		
3	2445		
4	2446		
5	2447		
6	2448		
7	2449		
8	2450		
9	2451		
10	2452		
11	2453		
12	2454		
13	2455		
Total rows: 620 of 620		Query complete 00:00:01.982	

Рис. 10 – SELECT 8

9. Составить таблицу, содержащую имя класса в формате буква_цифра, ФИО классного руководителя и ФИО старосты

```

SELECT class_name,
       year_start_date,
       year_end_date,
       student_full_name,
       headstudent_date_from,
       headstudent_date_to,
       teacher_full_name,
       headteacher_date_from,
       headteacher_date_to
FROM
  (SELECT concat(class_number::varchar, class_letter) AS class_name,
   year_start_date,
   year_end_date,
   class_id
  FROM "School_schema"."Curriculum"
  JOIN "School_schema"."Class" USING(curriculum_id)) AS TEMP1
LEFT JOIN
  (SELECT class_id,
   full_name AS student_full_name,
   "School_schema"."Headstudent".date_from AS headstudent_date_from,
   "School_schema"."Headstudent".date_to AS headstudent_date_to
  FROM "School_schema"."Curriculum"

```

```

JOIN "School_schema"."Class" USING(curriculum_id)
JOIN "School_schema"."Student_in_class" USING(class_id)
JOIN "School_schema"."Student" USING(student_code)
JOIN "School_schema"."Headstudent" USING(student_in_class_id)) AS TEMP2 USING(class_id)
LEFT JOIN
(SELECT class_id,
       full_name AS teacher_full_name,
       date_from AS headteacher_date_from,
       date_to AS headteacher_date_to
FROM "School_schema"."Curriculum"
JOIN "School_schema"."Class" USING(curriculum_id)
JOIN "School_schema"."Headteacher" USING(class_id)
JOIN "School_schema"."Teacher" USING(teacher_personnel_number)) AS TEMP3 USING(class_id)
ORDER BY year_start_date DESC,
         class_name ASC;

```

	class_name text	year_start_date date	year_end_date date	student_full_name character varying (250)	headstudent_date_from date	headstudent_date_to date	teacher_full_name character varying (250)
1	10A	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
2	10Б	2023-09-01	2024-08-30	[null]	[null]	[null]	Чаплин Тарас
3	11А	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
4	11Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
5	5А	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
6	5Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
7	5В	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
8	6А	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
9	6Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
10	6В	2023-09-01	2024-08-30	[null]	[null]	[null]	Норова Фелисити
11	7А	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
12	7Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
13	7В	2023-09-01	2024-08-30	[null]	[null]	[null]	Заборова Гоар
14	8А	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
15	8Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
16	8В	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
17	9А	2023-09-01	2024-08-30	[null]	[null]	[null]	Лорченкова Лиора
18	9Б	2023-09-01	2024-08-30	[null]	[null]	[null]	[null]
19	9В	2023-09-01	2024-08-30	[null]	[null]	[null]	Борн Тимур
20	10А	2022-09-01	2023-08-30	Астафьева Лара	2022-09-01	[null]	[null]
21	10Б	2022-09-01	2023-08-30	Собчак Артём	2022-09-01	[null]	Буров Мирослав

Total rows: 114 of 114 Query complete 00:00:00.047

Рис. 11 – SELECT 9

VIEW

1. Количество человек в классе (добавить поле в табличку Класс)

```
DROP VIEW IF EXISTS ClassNew;
```

```
CREATE VIEW ClassNew AS
```

```

(SELECT CONCAT(class_number::CHARACTER varying, ' ', class_letter) AS class_name,

```

```

        year_start_date,
        year_end_date,
        n_students_on_the_list,
        status
FROM "School_schema"."Class"
JOIN
    (SELECT class_id,
        COUNT(*) AS n_students_on_the_list
    FROM
        (SELECT *
        FROM "School_schema"."Class"
        JOIN "School_schema"."Student_in_class" USING(class_id)
        WHERE "School_schema"."Class".status = 'Расформирован'
        AND "School_schema"."Student_in_class".status = 'Числился'
        OR "School_schema"."Student_in_class".status = 'Числится') AS TEMP
1
        GROUP BY class_id) AS TEMP2 USING(class_id)
JOIN "School_schema"."Curriculum" USING(curriculum_id));

SELECT *
FROM ClassNew;

```

Query		Query History	
1	Data Output		Messages Explain × Notifications
2			
3			
4		class_name	year_start_date
5		text	date
6			year_end_date
7	1	10 Б	2020-09-01
8	2	7 Б	2019-09-01
9	3	6 А	2019-09-01
10	4	11 А	2020-09-01
11	5	5 А	2021-09-01
12	6	7 Б	2018-09-01
13	7	8 Б	2022-09-01
14	8	10 Б	2021-09-01
	9	10 А	2020-09-01
	10	8 А	2019-09-01
	11	9 Б	2021-09-01
	12	8 Б	2021-09-01
	13	6 А	2018-09-01
	14	9 В	2019-09-01
	15	9 А	2020-09-01
	16	9 А	2021-09-01
	17	9 Б	2020-09-01
	18	8 А	2021-09-01
	19	6 В	2021-09-01
	20	9 Б	2022-09-01
	21	8 А	2018-09-01
	22	10 А	2019-09-01
	23	7 А	2020-09-01
Total rows: 87 of 87		Query complete 00:00:00.073	

Рис. 12 – VIEW 1

2. Расчет общей нагрузки преподавателя (добавить поле в табличку Преподаватель)

```
DROP VIEW IF EXISTS TeacherWithHours;
```

```
CREATE VIEW TeacherWithHours AS
```

```
(SELECT *
```

```
FROM "School_schema"."Teacher"
```

```
JOIN
```

```
(SELECT teacher_personnel_number,
       date_from,
       date_to,
```

```
       SUM(hours) AS total_hours
```

```
FROM "School_schema"."Teacher"
```

```
JOIN "School_schema"."Workload" USING(teacher_personnel_number)
```

```

GROUP BY teacher_personnel_number,
         date_from,
         date_to) AS TEMP1 USING(teacher_personnel_number)

ORDER BY teacher_personnel_number,
         date_from,
         date_to);

SELECT *
FROM TeacherWithHours;

```

	teacher_personnel_number integer	full_name character varying (250)	birth_date date	passport character varying (11)	address character varying (250)	phone character varying (18)	email character varying (100)
1	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com
2	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com
3	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com
4	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com
5	0	Басманова Ильзира	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com
6	1	Клепов Геннадий	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com
7	1	Клепов Геннадий	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com
8	1	Клепов Геннадий	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com
9	1	Клепов Геннадий	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com
10	1	Клепов Геннадий	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com
11	2	Евлампиев Всеволод	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com
12	2	Евлампиев Всеволод	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com
13	2	Евлампиев Всеволод	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com
14	2	Евлампиев Всеволод	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com
15	2	Евлампиев Всеволод	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com
16	3	Яковлева Доминика	1961-02-08	7586 968243	[null]	+7 (638) 377-48-41	minimal1864@example.org
17	3	Яковлева Доминика	1961-02-08	7586 968243	[null]	+7 (638) 377-48-41	minimal1864@example.org

Total rows: 200 of 200 Query complete 00:00:00.060 Ln 9, Col 8

Рис. 13 – VIEW 2, левая часть таблицы

	birth_date date	passport character varying (11)	address character varying (250)	phone character varying (18)	email character varying (100)	date_from date	date_to date	total_hours bigint
1	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com	2018-09-01	2019-05-31	360
2	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com	2019-09-01	2020-05-31	324
3	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com	2020-09-01	2021-05-31	324
4	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com	2021-09-01	2022-05-31	288
5	1963-08-16	1272 604506	Россия, Ивановская область, Иваново, Аллея Басовская 327	+7 (919) 492-06-48	canadian1815@example.com	2022-09-01	2023-05-31	180
6	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com	2018-09-01	2019-05-31	360
7	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com	2019-09-01	2020-05-31	144
8	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com	2020-09-01	2021-05-31	324
9	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com	2021-09-01	2022-05-31	360
10	1995-03-12	8476 193077	[null]	+7 (641) 823-90-12	specially1872@yandex.com	2022-09-01	2023-05-31	288
11	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com	2018-09-01	2019-05-31	396
12	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com	2019-09-01	2020-05-31	180
13	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com	2020-09-01	2021-05-31	288
14	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com	2021-09-01	2022-05-31	324
15	1997-11-13	4690 722033	[null]	+7 (199) 297-51-20	lindsay2082@yahoo.com	2022-09-01	2023-05-31	144
16	1961-02-08	7586 968243	[null]	+7 (638) 377-48-41	minimal1864@example.org	2018-09-01	2019-05-31	360
17	1961-02-08	7586 968243	[null]	+7 (638) 377-48-41	minimal1864@example.org	2019-09-01	2020-05-31	180

Total rows: 200 of 200 Query complete 00:00:00.060 Ln 9, Col 9

Рис. 14 – VIEW 2, правая часть таблицы

INSERT, UPDATE, DELETE

INSERT делает следующее: для классов текущего года, у которых нет классных руководителей, добавляет классных руководителей из числа не имеющих классного руководства учителей. Действия производятся над таблицей «Классное руководство», так как именно она связывает классы и учителей.

Скрипт:

```
DROP VIEW IF EXISTS ClassesNoHeadteachers;

CREATE VIEW ClassesNoHeadteachers AS
(SELECT class_id
 FROM "School_schema"."Class" AS class_no_headteacher_id
 WHERE status = 'Действует'
 AND NOT EXISTS
   (SELECT class_id
    FROM
      (SELECT *
       FROM "School_schema"."Class"
       WHERE status = 'Действует') AS TEMP1
    JOIN "School_schema"."Headteacher" USING(class_id) AS TEMP2
    WHERE class_id = class_no_headteacher_id.class_id));

INSERT INTO "School_schema"."Headteacher" (date_from,
                                             date_to,
                                             teacher_personnel_number,
                                             class_id)

SELECT date_from,
       date_to,
       teacher_personnel_number,
       class_id
FROM
  (SELECT TEMP4.*,
   row_number() OVER () AS rnum
   FROM
     (SELECT teacher_personnel_number,
      date(now()) AS date_from,
      '30.08.2023'::date AS date_to
     FROM
       (SELECT teacher_personnel_number
        FROM "School_schema"."Teacher" AS teacher_no_headteacher_number
        WHERE NOT EXISTS
          (SELECT teacher_personnel_number
           FROM
             (SELECT *
              FROM
                (SELECT *
                 FROM "School_schema"."Class"
                 WHERE status = 'Действует') AS TEMP1
              JOIN "School_schema"."Headteacher" USING(class_id)
              JOIN "School_schema"."Teacher" USING(teacher_personnel_numbe
r)) AS TEMP2
            WHERE teacher_personnel_number = teacher_no_headteacher_number.
teacher_personnel_number)) AS TEMP3
```

```

ORDER BY RANDOM()
LIMIT
(SELECT COUNT(*)
FROM ClassesNoHeadteachers)) AS TEMP4) AS TEMP5
JOIN
(SELECT ClassesNoHeadteachers.*,
row_number() OVER () AS rnum
FROM ClassesNoHeadteachers) AS TEMP6 USING(rnum);

```

Query Query History

18 Data Output Messages Notifications

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

	headteacher_id [PK] integer	date_from date	date_to date	teacher_personnel_number integer	class_id integer
17	17	2021-09-01	2022-08-30	13	64
18	18	2021-09-01	2022-08-30	10	66
19	19	2022-09-01	2023-08-30	11	94
20	20	2019-09-01	2020-08-30	12	20
21	21	2018-09-01	2019-08-30	37	3
22	22	2022-09-01	2023-08-30	5	89
23	23	2022-09-01	2023-08-30	39	87
24	24	2023-09-01	[null]	9	110
25	25	2019-09-01	2020-08-30	29	22
26	26	2018-09-01	2019-08-30	2	17
27	27	2019-09-01	2020-08-30	15	23
28	28	2020-09-01	2021-08-30	31	39
29	29	2021-09-01	2022-08-30	1	59
30	30	2021-09-01	2022-08-30	34	73
31	31	2023-09-01	[null]	14	104
32	32	2018-09-01	2019-08-30	6	14
33	33	2019-09-01	2020-08-30	0	35
34	34	2022-09-01	2023-08-30	17	91
35	35	2023-09-01	[null]	4	101
36	36	2021-09-01	2022-08-30	30	65
37	37	2018-09-01	2019-08-30	3	2
38	38	2021-09-01	2022-08-30	19	62
39	39	2022-09-01	2023-08-30	7	93
40	40	2018-09-01	2019-08-30	32	7

Total rows: 40 of 40

Query complete 00:00:00.165

Рис. 15 – Таблица до INSERT-а

Query Query History

Data Output Messages Notifications

	headteacher_id [PK] integer	date_from date	date_to date	teacher_personnel_number integer	class_id integer
31	31	2023-09-01	[null]	14	104
32	32	2018-09-01	2019-08-30	6	14
33	33	2019-09-01	2020-08-30	0	35
34	34	2022-09-01	2023-08-30	17	91
35	35	2023-09-01	[null]	4	101
36	36	2021-09-01	2022-08-30	30	65
37	37	2018-09-01	2019-08-30	3	2
38	38	2021-09-01	2022-08-30	19	62
39	39	2022-09-01	2023-08-30	7	93
40	40	2018-09-01	2019-08-30	32	7
41	41	2023-03-19	2023-08-30	25	77
42	42	2023-03-19	2023-08-30	13	78
43	43	2023-03-19	2023-08-30	30	79
44	44	2023-03-19	2023-08-30	32	80
45	45	2023-03-19	2023-08-30	12	81
46	46	2023-03-19	2023-08-30	0	82
47	47	2023-03-19	2023-08-30	36	83
48	48	2023-03-19	2023-08-30	35	84
49	49	2023-03-19	2023-08-30	20	85
50	50	2023-03-19	2023-08-30	16	86
51	51	2023-03-19	2023-08-30	22	88
52	52	2023-03-19	2023-08-30	24	90
53	53	2023-03-19	2023-08-30	8	92
54	54	2023-03-19	2023-08-30	27	95

Total rows: 54 of 54 Query complete 00:00:00.092

Рис. 16 – Таблица после INSERT-а

UPDATE делает следующее: повышает на балл оценку всем тем, кто за К/р, С/р или Д/з получил тройку 27.10.2022.

Скрипт:

```
UPDATE "School_schema"."Grade"
SET grade = grade + 1,
    note = 'Акция невиданной щедрости'
WHERE grade_id IN
(SELECT grade_id
```



```

FROM "School_schema"."Lesson"
JOIN "School_schema"."Attendance" USING(lesson_id)
JOIN "School_schema"."Grade" USING(attendance_id)
JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id)
WHERE lesson_date = '27.10.2022'::date
      AND attendance
      AND grade = 3
      AND TYPE LIKE '%/%');

```

Query

Query History

1

2

3

4

5

6

7

8

9

```
SELECT lesson_date, grade_id, grade, type FROM "School_schema"."Lesson" JOIN "School_schema"."Attendance"
USING(lesson_id) JOIN "School_schema"."Grade" USING(attendance_id) JOIN "School_schema"."EvaluationTool"
USING(evaluation_tool_id) WHERE lesson_date = '27.10.2022'::date;
```

Data Output

Messages

Notifications

	lesson_date date	grade_id integer	grade integer	type character varying (20)
1	2022-10-27	48268	3	Иное
2	2022-10-27	48269	5	Д/з
3	2022-10-27	90480	3	Иное
4	2022-10-27	61423	5	К/р
5	2022-10-27	61424	4	Д/з
6	2022-10-27	90434	3	Работа на уроке
7	2022-10-27	74242	3	С/р
8	2022-10-27	90481	3	Иное
9	2022-10-27	90482	3	С/р
10	2022-10-27	291161	3	Ведение тетради
11	2022-10-27	128642	3	К/р
12	2022-10-27	174528	3	К/р
13	2022-10-27	174529	4	Ведение тетради
14	2022-10-27	191710	4	С/р
15	2022-10-27	291211	5	Ведение тетради
16	2022-10-27	159713	3	Иное
17	2022-10-27	159714	5	Ведение тетради
18	2022-10-27	159750	3	Ведение тетради
19	2022-10-27	159751	3	Иное
20	2022-10-27	159752	4	Ведение тетради
21	2022-10-27	159753	5	Иное

Total rows: 50 of 50

Query complete 00:00:00.275

Рис. 17 – Таблица до UPDATE-а

Query

Query History

```
1 SELECT lesson_date, grade_id, grade, type, "School_schema"."Grade".note FROM "School_schema"."Lesson"
2 JOIN "School_schema"."Attendance" USING(lesson_id) JOIN "School_schema"."Grade" USING(attendance_id)
3 JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id) WHERE lesson_date = '27.10.2022'::date
4 ORDER BY grade_id;
```

Data Output

Messages

Notifications

	lesson_date date	grade_id integer	grade integer	type character varying (20)	note text
1	2022-10-27	48268	3	Иное	[null]
2	2022-10-27	48269	5	Д/з	[null]
3	2022-10-27	61423	5	К/р	[null]
4	2022-10-27	61424	4	Д/з	[null]
5	2022-10-27	74242	4	С/р	Акция невиданной щедрости
6	2022-10-27	90434	3	Работа на уроке	[null]
7	2022-10-27	90480	3	Иное	[null]
8	2022-10-27	90481	3	Иное	[null]
9	2022-10-27	90482	4	С/р	Акция невиданной щедрости
10	2022-10-27	128642	4	К/р	Акция невиданной щедрости
11	2022-10-27	159713	3	Иное	[null]
12	2022-10-27	159714	5	Ведение тетради	[null]
13	2022-10-27	159750	3	Ведение тетради	[null]
14	2022-10-27	159751	3	Иное	[null]
15	2022-10-27	159752	4	Ведение тетради	[null]
16	2022-10-27	159753	5	Иное	[null]
17	2022-10-27	174528	4	К/р	Акция невиданной щедрости
18	2022-10-27	174529	4	Ведение тетради	[null]
19	2022-10-27	191710	4	С/р	[null]
20	2022-10-27	291161	3	Ведение тетради	[null]

Total rows: 50 of 50

Query complete 00:00:00.235

Рис. 18 – Таблица после UPDATE-а

DELETE делает следующее: удаляет из таблички «Ученик» всех ребят, которые не являются обучающимися школы.

Скрипт:

```

DELETE
FROM "School_schema"."Student"
WHERE student_code NOT IN
  (SELECT student_code
   FROM "School_schema"."Student" AS TEMP1
   WHERE EXISTS
     (SELECT student_code
      FROM "School_schema"."Student_in_class" AS TEMP2
      WHERE TEMP1.student_code = TEMP2.student_code));

```

Query

Query History

1 SELECT * FROM "School_schema"."Student";

2

Data Output

Messages

Notifications

	student_code [PK] integer	full_name character varying (250)	birth_date date	address character varying (250)	phone character va
1	1	Шахмагонов Созон	2002-07-27	Россия, Ивановская область, Иваново, Аллея Шереметьевская 545	[null]
2	2	Шарапова Замира	2002-03-09	[null]	[null]
3	3	Гранин Ермолай	2002-05-06	[null]	[null]
4	4	Вавилова Юлия	2002-08-17	Россия, Ивановская область, Иваново, Аллея Заозерная 149	[null]
5	5	Воробьева Ивета	2002-09-09	Россия, Ивановская область, Иваново, ул. Очаковская 518	[null]
6	6	Новгородцева Алексия	2002-06-18	Россия, Ивановская область, Иваново, ул. Ржевский М. 1034	+7 (252) 82
7	7	Свитнев Вячеслав	2002-12-15	Россия, Ивановская область, Иваново, Аллея Открытое 117	[null]
8	8	Тесаков Хрисанф	2002-09-17	Россия, Ивановская область, Иваново, ул. Академика Петровского 430	[null]
9	9	Иванов Созон	2002-11-02	Россия, Ивановская область, Иваново, ул. Ивановская 930	[null]
10	10	Белокурова Алла	2002-01-22	Россия, Ивановская область, Иваново, ул. Рыбинская 25	+7 (978) 17
11	11	Юдин Венедикт	2002-10-14	[null]	+7 (826) 65
12	12	Мозговой Иосиф	2002-08-15	Россия, Ивановская область, Иваново, Аллея Пересветов 998	+7 (332) 09
13	13	Тетерин Никанор	2002-04-19	[null]	[null]
14	14	Маркелова Тина	2002-09-01	[null]	[null]
15	15	Малинина Гоар	2002-09-17	Россия, Ивановская область, Иваново, Аллея Тарутинская 157	[null]
16	16	Фетисова Айсун	2002-03-15	Россия, Ивановская область, Иваново, Аллея Дмитровская 878	+7 (131) 83
17	17	Хлумов Варлам	2002-07-24	Россия, Ивановская область, Иваново, ул. Усиевича 909	[null]
18	18	Старикова Алеста	2002-02-24	Россия, Ивановская область, Иваново, Аллея Пушкина 264	+7 (495) 26
19	19	Гранина Домника	2002-05-28	[null]	[null]
20	20	Евстифеева Мелания	2002-09-15	[null]	[null]
21	21	Петрушевский Лукьян	2002-11-08	Россия, Ивановская область, Иваново, Аллея Оленья М. 11	[null]
22	22	Попова Роберта	2002-12-09	[null]	+7 (211) 36

Total rows: 1000 of 5000

Query complete 00:00:00.083

Рис. 19 – Таблица до DELETE-а

Query

Query History

1 SELECT * FROM "School_schema"."Student";

Data Output

Messages

Notifications

	student_code [PK] integer	full_name character varying (250)	birth_date date	address character varying (250)	phone character
1	1	Шахмагонов Созон	2002-07-27	Россия, Ивановская область, Иваново, Аллея Шереметьевская 545	[null]
2	2	Шарапова Замира	2002-03-09	[null]	[null]
3	3	Гранин Ермолай	2002-05-06	[null]	[null]
4	4	Вавилова Юлия	2002-08-17	Россия, Ивановская область, Иваново, Аллея Заозерная 149	[null]
5	5	Воробьева Ивета	2002-09-09	Россия, Ивановская область, Иваново, ул. Очаковская 518	[null]
6	6	Новгородцева Алексия	2002-06-18	Россия, Ивановская область, Иваново, ул. Ржевский М. 1034	+7 (252)
7	7	Свитнев Вячеслав	2002-12-15	Россия, Ивановская область, Иваново, Аллея Открытое 117	[null]
8	8	Тесаков Хрисанф	2002-09-17	Россия, Ивановская область, Иваново, ул. Академика Петровского 430	[null]
9	9	Иванов Созон	2002-11-02	Россия, Ивановская область, Иваново, ул. Ивановская 930	[null]
10	10	Белокурова Алла	2002-01-22	Россия, Ивановская область, Иваново, ул. Рыбинская 25	+7 (978)
11	11	Юдин Венедикт	2002-10-14	[null]	+7 (826)
12	12	Мозговой Иосиф	2002-08-15	Россия, Ивановская область, Иваново, Аллея Пересветов 998	+7 (332)
13	13	Тетерин Никанор	2002-04-19	[null]	[null]
14	14	Маркелова Тина	2002-09-01	[null]	[null]
15	15	Малинина Гоар	2002-09-17	Россия, Ивановская область, Иваново, Аллея Тарутинская 157	[null]
16	16	Фетисова Айсун	2002-03-15	Россия, Ивановская область, Иваново, Аллея Дмитровская 878	+7 (131)
17	17	Хлумов Варлам	2002-07-24	Россия, Ивановская область, Иваново, ул. Усиевича 909	[null]
18	18	Старикова Алеста	2002-02-24	Россия, Ивановская область, Иваново, Аллея Пушкина 264	+7 (495)
19	19	Гранина Домника	2002-05-28	[null]	[null]
20	20	Евстифеева Мелания	2002-09-15	[null]	[null]
21	21	Петрушевский Лукьян	2002-11-08	Россия, Ивановская область, Иваново, Аллея Оленья М. 11	[null]
22	22	Попова Роберта	2002-12-09	[null]	+7 (211)

Total rows: 1000 of 1224 Query complete 00:00:00.071

Рис. 20 – Таблица после DELETE-а

Небольшой эксперимент.

Во время написания SELECT-ов с JOIN-ами я задалась вопросом, эффективнее ли фильтровать по условию между JOIN-ами или разом в конце. Решила взять запрос, аналогичный использованному в UPDATE.

Запрос 1:

```
SELECT grade_id
FROM "School_schema"."Lesson"
JOIN "School_schema"."Attendance" USING(lesson_id)
JOIN "School_schema"."Grade" USING(attendance_id)
JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id)
WHERE lesson_date = '27.10.2022'::date
AND attendance
AND grade = 5
AND TYPE LIKE '%/%';
```

Запрос 2:

```
SELECT grade_id
FROM
  (SELECT *
   FROM
     (SELECT *
      FROM "School_schema"."Lesson"
      WHERE lesson_date = '27.10.2022'::date) AS TEMP1
     JOIN "School_schema"."Attendance" USING(lesson_id)
     WHERE attendance) AS TEMP2
    JOIN "School_schema"."Grade" USING(attendance_id)
    WHERE grade = 5) AS TEMP3
JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id)
WHERE TYPE LIKE '%/%';
```

```
from scipy.stats import ttest_rel
```

```
query1 = [166, 200, 255, 194, 227, 224, 241, 223, 200, 158, 193, 175, 208, 272, 266, 147, 268, 241, 178, 211]
query2 = [149, 160, 192, 244, 201, 251, 163, 209, 195, 178, 258, 181, 195, 273, 273, 205, 210, 255, 227, 257]
ttest_rel(query1, query2).pvalue
```

```
0.8767765073024911
```

Рис. 21 – Значение pvalue для парного t-теста двух выборок по 20 наблюдений

Вывод: нулевую гипотезу о равенстве средних нельзя отклонить, а значит существенной разницы, как именно строить запрос (по крайней мере в данном случае), нет.

INDEX

Простой индекс.

Query History

1 EXPLAIN SELECT full_name, subject_name, SUM(hours) AS hours FROM ((SELECT * FROM "School_schema"."Teacher"

2 JOIN "School_schema"."Workload" USING(teacher_personnel_number) WHERE '01.09.2022'>= date_from AND

3 date_from <= '31.08.2023') AS TEMP1 JOIN (SELECT discipline_id, subject_name FROM "School_schema"."Discipline")

4 AS TEMP2 USING(discipline_id)) GROUP BY teacher_personnel_number, full_name, discipline_id, subject_name

5 ORDER BY hours;

6

7

Data Output Messages Explain x Notifications

+

📄

▼

🗑️

📦

⬇️

📈

QUERY PLAN

text

1 Sort (cost=194.51..198.64 rows=1652 width=555)

2 Sort Key: (sum("Workload".hours))

3 -> HashAggregate (cost=89.69..106.21 rows=1652 width=555)

4 Group Key: "Teacher".teacher_personnel_number, "Workload".discipline_id, "Discipline".subj...

5 -> Hash Join (cost=26.30..73.17 rows=1652 width=551)

6 Hash Cond: ("Workload".discipline_id = "Discipline".discipline_id)

7 -> Hash Join (cost=11.13..53.61 rows=1652 width=528)

8 Hash Cond: ("Workload".teacher_personnel_number = "Teacher".teacher_personnel_number)

9 -> Seq Scan on "Workload" (cost=0.00..37.78 rows=1652 width=12)

10 Filter: (("2022-09-01"::date >= date_from) AND (date_from <= '2023-08-31'::date))

11 -> Hash (cost=10.50..10.50 rows=50 width=520)

12 -> Seq Scan on "Teacher" (cost=0.00..10.50 rows=50 width=520)

13 -> Hash (cost=10.08..10.08 rows=408 width=27)

14 -> Seq Scan on "Discipline" (cost=0.00..10.08 rows=408 width=27)

total rows: 14 of 14

Query complete 00:00:00.096

✓ Successfully run. Total

Рис. 22 – План выполнения запроса без индексов

Query History

```

1 CREATE INDEX date_index ON "School_schema"."Workload"(date_from);
2
3 EXPLAIN SELECT full_name, subject_name, SUM(hours) AS hours FROM ((SELECT * FROM "School_schema"."Teacher"
4 JOIN "School_schema"."Workload" USING(teacher_personnel_number) WHERE '01.09.2022'>= date_from AND
5 date_from <= '31.08.2023') AS TEMP1 JOIN (SELECT discipline_id, subject_name FROM "School_schema"."Discipline")
6 AS TEMP2 USING(discipline_id)) GROUP BY teacher_personnel_number, full_name, discipline_id, subject_name
7 ORDER BY hours;

```

Data Output Messages Explain x Notifications

+

📄

▼

🗑️

📦

⬇️

📈

	QUERY PLAN	
	text	🔒
1	Sort (cost=194.51..198.64 rows=1652 width=555)	
2	Sort Key: (sum("Workload".hours))	
3	-> HashAggregate (cost=89.69..106.21 rows=1652 width=555)	
4	Group Key: "Teacher".teacher_personnel_number, "Workload".discipline_id, "Discipline".subj...	
5	-> Hash Join (cost=26.30..73.17 rows=1652 width=551)	
6	Hash Cond: ("Workload".discipline_id = "Discipline".discipline_id)	
7	-> Hash Join (cost=11.13..53.61 rows=1652 width=528)	
8	Hash Cond: ("Workload".teacher_personnel_number = "Teacher".teacher_personnel_number)	
9	-> Seq Scan on "Workload" (cost=0.00..37.78 rows=1652 width=12)	
10	Filter: ((2022-09-01::date >= date_from) AND (date_from <= 2023-08-31::date))	
11	-> Hash (cost=10.50..10.50 rows=50 width=520)	
12	-> Seq Scan on "Teacher" (cost=0.00..10.50 rows=50 width=520)	
13	-> Hash (cost=10.08..10.08 rows=408 width=27)	
14	-> Seq Scan on "Discipline" (cost=0.00..10.08 rows=408 width=27)	

total rows: 14 of 14

Query complete 00:00:00.048

Рис. 23 – План выполнения запроса с индексами

Чтобы оценить влияние индексов на время выполнения запроса, был проведен парный t-тест для выборок по 20 наблюдений, где первая выборка – время выполнения запроса без индексов в миллисекундах, вторая – время выполнения запроса с индексами в миллисекундах. Полученное значение статистики не позволяет отклонить нулевую гипотезу о равенстве средних.

```

from scipy.stats import ttest_rel

query1 = [106, 136, 106, 120, 106, 157, 163, 78, 102, 59, 66, 61, 60, 78, 106, 74, 94, 164, 97, 55]
query2 = [66, 108, 119, 96, 183, 80, 109, 98, 64, 74, 93, 97, 104, 55, 70, 83, 85, 86, 104, 90]
ttest_rel(query1, query2).pvalue

```

0.5107217070749875

Рис. 24 – Значение pvalue для парного t-теста двух выборок по 20 наблюдений

Составной индекс.

Query History

```
1 EXPLAIN SELECT grade_id FROM "School_schema"."Lesson" JOIN "School_schema"."Attendance" USING(lesson_id) JOIN
2 "School_schema"."Grade" USING(attendance_id) JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id)
3 WHERE lesson_date = '27.10.2022'::date AND attendance AND grade = 5 AND type LIKE '%/%';
```

5
6
7
8

Data OutputMessagesExplain ×Notifications

QUERY PLAN

text

1

Gather (cost=19522.80..26673.60 rows=53 width=4)

2

Workers Planned: 2

3

-> Nested Loop (cost=18522.80..25668.30 rows=22 width=4)

4

-> Parallel Hash Join (cost=18522.51..25653.41 rows=44 width=8)

5

Hash Cond: ("Grade".attendance_id = "Attendance".attendance_id)

6

-> Parallel Seq Scan on "Grade" (cost=0.00..6797.11 rows=88963 width=12)

7

Filter: (grade = 5)

8

-> Parallel Hash (cost=18518.13..18518.13 rows=350 width=4)

9

-> Hash Join (cost=1178.24..18518.13 rows=350 width=4)

10

Hash Cond: ("Attendance".lesson_id = "Lesson".lesson_id)

11

-> Parallel Seq Scan on "Attendance" (cost=0.00..16397.65 rows=358929 width=8)

12

Filter: attendance

13

-> Hash (cost=1177.44..1177.44 rows=64 width=4)

14

-> Seq Scan on "Lesson" (cost=0.00..1177.44 rows=64 width=4)

15

Filter: (lesson_date = '2022-10-27'::date)

16

-> Index Scan using "EvaluationTool_pkey" on "EvaluationTool" (cost=0.29..0.34 rows=1 width=4)

17

Index Cond: (evaluation_tool_id = "Grade".evaluation_tool_id)

18

Filter: ((type)::text ~~ '%/%'::text)

d, attendance);

total rows: 18 of 18Query complete 00:00:00.064

Рис. 25 – План выполнения запроса без индексов

Query History

1

CREATE INDEX idx_lesson ON "School_schema"."Lesson"(lesson_id, lesson_date);

2

CREATE INDEX idx_attendance ON "School_schema"."Attendance"(lesson_id, attendance_id, attendance);

3

CREATE INDEX idx_grade ON "School_schema"."Grade"(attendance_id, grade_id, grade);

4

5

EXPLAIN SELECT grade_id FROM "School_schema"."Lesson" JOIN "School_schema"."Attendance" USING(lesson_id) JOIN

6

"School_schema"."Grade" USING(attendance_id) JOIN "School_schema"."EvaluationTool" USING(evaluation_tool_id)

7

WHERE lesson_date = '27.10.2022'::date AND attendance AND grade = 5 AND type LIKE '%/%';

8

Data Output Messages Explain X Notifications

+

📄

▼

🗑️

🔄

⬇️

📈

QUERY PLAN

text

1

Nested Loop (cost=1.15..1950.52 rows=53 width=4)

2

-> Nested Loop (cost=0.85..1914.98 rows=105 width=8)

3

-> Nested Loop (cost=0.43..1501.76 rows=839 width=4)

4

-> Seq Scan on "Lesson" (cost=0.00..1177.44 rows=64 width=4)

5

Filter: (lesson_date = '2022-10-27'::date)

6

-> Index Only Scan using idx_attendance on "Attendance" (cost=0.43..4.91 rows=16 width=8)

7

Index Cond: ((lesson_id = "Lesson".lesson_id) AND (attendance = true))

8

-> Index Scan using idx_grade on "Grade" (cost=0.42..0.48 rows=1 width=12)

9

Index Cond: ((attendance_id = "Attendance".attendance_id) AND (grade = 5))

10

-> Index Scan using "EvaluationTool_pkey" on "EvaluationTool" (cost=0.29..0.34 rows=1 width=4)

11

Index Cond: (evaluation_tool_id = "Grade".evaluation_tool_id)

12

Filter: ((type)::text ~~ '%/%':text)

total rows: 12 of 12

Query complete 00:00:01.354

Рис. 26 – План выполнения запроса с индексами

Чтобы оценить влияние индексов на время выполнения запроса, был проведен парный t-тест для выборок по 20 наблюдений, где первая выборка – время выполнения запроса без индексов в миллисекундах, вторая – время выполнения запроса с индексами в миллисекундах. Полученное значение статистики позволяет отклонить нулевую гипотезу о равенстве средних. Запрос с индексами выполняется быстрее.

```

from scipy.stats import ttest_rel

query1 = [166, 200, 255, 194, 227, 224, 241, 223, 200, 158, 193, 175, 208, 272, 266, 147, 268, 241, 178, 211]
query2 = [38, 42, 73, 46, 41, 40, 58, 54, 101, 38, 42, 39, 42, 59, 81, 119, 60, 69, 108, 58]
ttest_rel(query1, query2).pvalue

6.430758681717733e-12

```

Рис. 27 – Значение pvalue для парного t-теста двух выборок по 20 наблюдений

Выводы:

Первое, о чем необходимо сказать, это генерация данных для БД. На этапе создания база была заполнена малым количеством записей вручную. В процессе выполнении лабораторной работы стало понятно, что этого

количества данных не хватит для демонстрации работоспособности запросов. Для генерации данных было решено написать скрипт на языке программирования Python. Скрипт получился достаточно сложным, так как содержал большое количество условий. Например то, что один и тот же преподаватель не может вести урок в разных кабинетах одновременно; ученик не может получить оценку, если не присутствовал на уроке; физкультура не может проводиться в кабинете физики; магистратура не может быть закончена раньше бакалавриата; ребенок в 12 лет не может учиться в 11 классе и т. п. Скрипт использует такие библиотеки, как sqlalchemy (работа с базой данных), mimesis (генерация рандомных данных) и pandas (работа с таблицами в Python), и работает около 20 минут. По окончании генерации данных самая объемная таблица базы («Присутствие на уроке») содержала 1713036 строк.

После этапа заполнения базы данных большим количеством записей были созданы запросы и представления на выборку данных к базе данных PostgreSQL согласно индивидуальному заданию. Считаю важным отметить следующее: запросы, содержащие большое количество JOIN-ов, работают достаточно долго даже при наличии индексов (первичные и внешние ключи по умолчанию проиндексированы). Вероятно, это связано не столько с самими JOIN-ами, сколько с объемом соединяемых таблиц. Кроме того, нет особой разницы, проводить фильтрацию по условию последовательно после каждого JOIN-а или в конце после всех. Еще одно эмпирическое наблюдение, сделанное в ходе выполнения первой части задания: схема «select * from список_значений where список_значений.значение = статистика(список_значений.значение)» является стандартной для отбора строк таблицы, содержащих максимум и минимум.

В рамках данной лабораторной работы были составлены 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. При выполнении было изучено графическое представление запросов и просмотрена история запросов.

Наконец, были созданы простой и составной индексы для двух произвольных запросов. Сравнение времени выполнения запросов без индексов и с индексами показало, что запросы с индексами выполняются быстрее в случае, если индексируемые данные выбраны корректно. Так, в моем случае использование единичного простого индекса не дало прироста в производительности запроса, но при этом использование совокупности составных оказало положительный эффект. Для сравнения времени запросов использовался парный t-тест из библиотеки scipy.stats на Python. Для получения плана запроса была использована команда EXPLAIN, показавшая, что запросы с индексами содержат меньшее количество операций, если индексируемые значения выбраны корректно. При удалении индексов возникла проблема: индекс существовал, но при этом вылетала ошибка, что

его не существует. Оказалось, что если таблицы лежат не в public схеме, то перед удалением индексов необходимо прописать SET search_path = <schema_name>.

Форматирование кода было произведено с помощью инструмента sqlformat.org.