

Санкт-Петербургский 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 баз данных»

Автор:

Ивенкова Е.Д.

группа К32422

Преподаватель:

Говорова М.М.

Санкт-Петербург

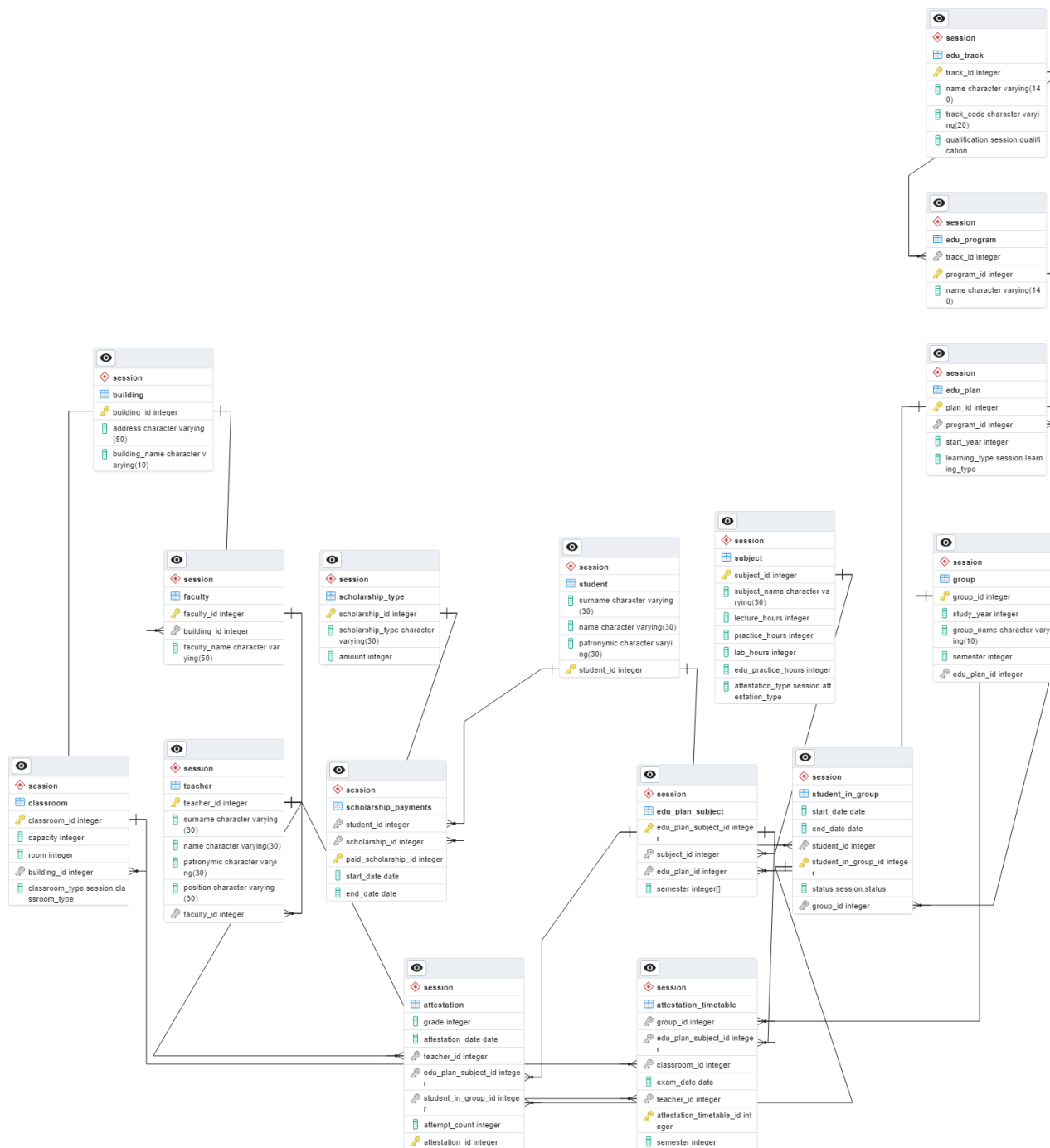
2023

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

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:



Индивидуальное задание:

Вариант 2. БД «Сессия»

Задание 2. Создайте запросы:

- 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.
- 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.
- Вывести список студентов, сдавших все положенные экзамены.
- Вывести список студентов, получивших максимальный средний балл в своей группе.

Задание 3. Создайте 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).

Выполнение:

Задание 2

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

```
select
session.teacher.surname,
session.attestation_timetable.exam_date,
session.subject.subject_name
from
session.attestation_timetable
inner join session.teacher
    on session.teacher.teacher_id =
        session.attestation_timetable.teacher_id
inner join session.edu_plan_subject
    on session.edu_plan_subject.edu_plan_subject_id =
        session.attestation_timetable.edu_plan_subject_id
inner join session.subject
    on session.edu_plan_subject.subject_id =
        session.subject.subject_id
where
session.attestation_timetable.group_id in

(select group_id from session.group
where group_name = 'K3218')
```

Query Query History

```
1 select
2     session.teacher.surname,
3     session.attestation_timetable.exam_date,
4     session.subject.subject_name
5 from
6     session.attestation_timetable
7     inner join session.teacher
8         on session.teacher.teacher_id =
9             session.attestation_timetable.teacher_id
10    inner join session.edu_plan_subject
11        on session.edu_plan_subject.edu_plan_subject_id =
12            session.attestation_timetable.edu_plan_subject_id
13    inner join session.subject
14        on session.edu_plan_subject.subject_id =
15            session.subject.subject_id
16 where
17     session.attestation_timetable.group_id in
18
19     (select group_id from session.group
20      where group_name = 'K3218')
21
```

Data Output Messages Notifications

	surname character varying (30) 🔒	exam_date date 🔒	subject_name character varying (30) 🔒
1	Леонов	2018-09-18	Общая физика
2	Соловьёва	2018-09-29	Дискретная математика
3	Филиппов	2019-01-21	Дискретная математика
4	Жданова	2019-01-18	Теория вероятностей
5	Жданова	2018-09-24	Средства веб-программирования

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

```
select
session.student.surname as student_surname,
session.student.name as student_name,
session.teacher.surname as teacher_surname
from
session.attestation
inner join session.student_in_group
    on session.attestation.student_in_group_id =
        session.student_in_group.student_in_group_id
inner join session.student
    on session.student.student_id =
        session.student_in_group.student_id
inner join session.teacher
    on session.teacher.teacher_id =
        session.attestation.teacher_id
where session.attestation.attempt_count = 1
and session.attestation.grade < 60
```

Query Query History

```

1 select
2     session.student.surname as student_surname,
3     session.student.name as student_name,
4     session.teacher.surname as teacher_surname
5 from
6     session.attestation
7     inner join session.student_in_group
8         on session.attestation.student_in_group_id =
9             session.student_in_group.student_in_group_id
10    inner join session.student
11        on session.student.student_id =
12            session.student_in_group.student_id
13    inner join session.teacher
14        on session.teacher.teacher_id =
15            session.attestation.teacher_id
16 where
17     session.attestation.attempt_count = 1
18     and
19     session.attestation.grade < 60
20
21

```

Data Output Messages Notifications

	student_surname character varying (30)	student_name character varying (30)	teacher_surname character varying (30)
1	Малышев	Егор	Павлова
2	Малышев	Егор	Волкова
3	Котов	Олег	Соловьёва
4	Смирнов	Олег	Павлова
5	Смирнов	Олег	Соловьёва
6	Смирнов	Олег	Волкова
7	Плотникова	Анна	Павлова
8	Смирнова	Елена	Соловьёва
9	Смирнова	Елена	Волкова
10	Соловьёва	Ангелина	Павлова
Total rows: 484 of 484 Query complete 00:00:00.070			

Покажем, что повтор имен студентов = разные предметы.

Query Query History

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

```
select
  session.student.surname as student_surname,
  session.student.name as student_name,
  session.teacher.surname as teacher_surname,
  session.attestation.edu_plan_subject_id
from
  session.attestation
  inner join session.student_in_group
    on session.attestation.student_in_group_id =
      session.student_in_group.student_in_group_id
  inner join session.student
    on session.student.student_id =
      session.student_in_group.student_id
  inner join session.teacher
    on session.teacher.teacher_id =
      session.attestation.teacher_id
where
  session.attestation.attempt_count = 1
  and
  session.attestation.grade < 60
```

Data Output Messages Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	student_surname character varying (30) 🔒	student_name character varying (30) 🔒	teacher_surname character varying (30) 🔒	edu_plan_subject_id integer 🔒
1	Малышев	Егор	Павлова	41
2	Малышев	Егор	Волкова	15
3	Котов	Олег	Соловьёва	18
4	Смирнов	Олег	Павлова	41
5	Смирнов	Олег	Соловьёва	18
6	Смирнов	Олег	Волкова	15
7	Плотникова	Анна	Павлова	41
8	Смирнова	Елена	Соловьёва	18
9	Смирнова	Елена	Волкова	15
10	Соловьёва	Ангелина	Павлова	41

Total rows: 484 of 484 Query complete 00:00:00.059

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

```
select distinct session.student.surname from
session.attestation
    inner join session.student_in_group
        on session.attestation.student_in_group_id
            = session.student_in_group.student_in_group_id
    inner join session.student
        on session.student_in_group.student_id
            = session.student.student_id
where
session.attestation.grade >
    (select avg(grade) from session.attestation
    where
        -- убираем должников
        attempt_count = 1 and grade >= 60
        and edu_plan_subject_id = 4)
and edu_plan_subject_id = 4
and attempt_count = 1
```

Query Query History



```
1 select distinct session.student.surname from
2     session.attestation
3     inner join session.student_in_group
4         on session.attestation.student_in_group_id
5             = session.student_in_group.student_in_group_id
6     inner join session.student
7         on session.student_in_group.student_id
8             = session.student.student_id
9 where
10     session.attestation.grade >
11         (select avg(grade) from session.attestation
12         where
13             -- убираем должников
14             attempt_count = 1 and grade >= 60
15             and edu_plan_subject_id = 4)
16 and edu_plan_subject_id = 4
17 and attempt_count = 1
18
```

Data Output Messages Explain × Notifications



	surname character varying (30)
1	Жданова
2	Лебедев
3	Леонов
4	Павлова
5	Сидоров
6	Смирнов
7	Соловьёва

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

Plot twist: было сгенерировано ОЧЕНЬ МАЛО ДАННЫХ, в связи с чем сделаем рейтинг просто какой-то одной сессии (например, посмотрим на средний балл групп после сдачи зимней сессии 2018 года)

```
select session.group.group_name from session.group
inner join session.student_in_group
on session.group.group_id = session.student_in_group.group_id
inner join session.attestation
on session.attestation.student_in_group_id =
session.student_in_group.student_in_group_id

where
attestation_date > '2018-12-31' and attestation_date < '2019-
02-01'
group by
session.group.group_id
order by
avg(grade) desc
```

Query Query History

```
1 select session.group.group_name from session.group
2 inner join session.student_in_group
3   on session.group.group_id = session.student_in_group.group_id
4 inner join session.attestation
5   on session.attestation.student_in_group_id = session.student_in_group.student_in_group_id
6
7 where
8   attestation_date > '2018-12-31' and attestation_date < '2019-02-01'
9 group by
10  session.group.group_id
11 order by
12  avg(grade) desc
13
14
15
16
17
```

Data Output Messages Notifications

group_name
character varying (10) 🔒

1	R3214
2	V3220
3	K3218
4	M3213

В идеале, если бы данных было достаточно, то выбор соответствующих групп было бы проще реализовать вложенными запросами. Тогда бы текст запроса выглядел так:

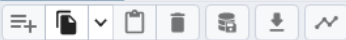
```
select session.group.group_name from session.group
inner join session.student_in_group
on session.group.group_id = session.student_in_group.group_id
inner join session.attestation
on session.attestation.student_in_group_id =
session.student_in_group.student_in_group_id
where
attestation_date > '2018-12-31' and attestation_date < '2019-
02-01'
and
session.group.group_id in
(select group_id from session.group where edu_plan_id in
(select plan_id from session.edu_plan
where
    plan_id in
    (select plan_id from session.edu_plan
    where
        program_id in (select program_id from
session.edu_program
                                where
                                    program_id in
                                    (select track_id from
session.edu_track where name = 'Инфохимия')
                                )
        )
    )
)
)
)
group by
session.group.group_id
order by
avg(grade) desc
```

Но результат пустой:

Query Query History

```
1 select session.group.group_name from session.group
2 inner join session.student_in_group
3     on session.group.group_id = session.student_in_group.group_id
4 inner join session.attestation
5     on session.attestation.student_in_group_id = session.student_in_group.student_in_group_id
6 where
7     attestation_date > '2018-12-31' and attestation_date < '2019-02-01'
8 and
9     session.group.group_id in
10 (select group_id from session.group where edu_plan_id in
11     ((select plan_id from session.edu_plan
12         where
13             plan_id in
14             (select plan_id from session.edu_plan
15                 where
16                     program_id in (select program_id from session.edu_program
17                         where
18                             program_id in
19                             (select track_id from session.edu_track where name = 'Инфохимия')
20                     )
21             )
22         )
23 )
24 group by
25     session.group.group_id
26 order by
27     avg(grade) desc
```

Data Output Messages Notifications



group_name
character varying (10)

5. 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.

Сдать сессию без троек = 1 попытка, балл > 74- 1

Сдать с тройкой = балл > 60, ≤ 74, 1 попытка - 0

Не сдать – 2 и 3 попытки есть – 0

Выбираем семестр обучения: 3

```
select
session.student.surname,
session.group.group_name,
tmp.avg_grade,
tmp.scholarship_given
from
--id студента в группе, его балл и стипендия
(select
    attestation.student_in_group_id,
    avg(grade) as avg_grade,

case
    when attestation.student_in_group_id in
        -- список студентов, сдавших с тройками
        -- или не сдавших: остальные получают стипендию:
        (select distinct
session.attestation.student_in_group_id
    from session.attestation
    inner join session.student_in_group
        on
session.student_in_group.student_in_group_id =
        session.attestation.student_in_group_id

        where session.student_in_group.group_id in
            -- список нужных для проверки групп
            (select distinct group_id
from session.attestation_timetable
where semester = 3)
        and
            attempt_count > 1)
    then 0

    when attestation.student_in_group_id in
        (select distinct
session.attestation.student_in_group_id
```

```

        from session.attestation
        inner join session.student_in_group
            on
session.student_in_group.student_in_group_id =
            session.attestation.student_in_group_id

        where session.student_in_group.group_id in
            (select distinct group_id
             from session.attestation_timetable
             where semester = 3))

        and attestation.student_in_group_id not in
            (select distinct
session.attestation.student_in_group_id
             from session.attestation
             inner join session.student_in_group
                 on
session.student_in_group.student_in_group_id =
                    session.attestation.student_in_group_id

             where session.student_in_group.group_id in
                 -- список нужных для проверки групп
                 (select distinct group_id
                  from session.attestation_timetable
                  where semester = 3)
                 and
                 attempt_count > 1)
    then 1
    end scholarship_given
from
    session.attestation
group by attestation.student_in_group_id) as tmp

inner join session.student_in_group
    on session.student_in_group.student_in_group_id =
        tmp.student_in_group_id
inner join session.student
    on session.student.student_id =
        session.student_in_group.student_id
inner join session.group
    on session.group.group_id =
        session.student_in_group.group_id
where
semester = 3

order by
session.group.group_name,

```


session.student.surname

Query

Query History

55

and

56

attempt_count > 1)

57

then 1

58

end scholarship_given

59

from

60

session.attestation

61

group by attestation.student_in_group_id) as tmp

62

inner join session.student_in_group

63

on session.student_in_group.student_in_group_id =

64

tmp.student_in_group_id

65

inner join session.student

66

on session.student.student_id =

67

session.student_in_group.student_id

68

inner join session.group

69

on session.group.group_id =

70

Data Output

Messages

Notifications

≡+

	surname character varying (30)	group_name character varying (10)	avg_grade numeric	scholarship_given integer
1	Виноградов	K3218	78.66666666666667	1
2	Виноградов	K3218	80.00000000000000	1
3	Волкова	K3218	65.20000000000000	0
4	Гончарова	K3218	81.33333333333333	1
5	Жданова	K3218	64.60000000000000	0
6	Жданова	K3218	83.66666666666667	1
7	Котов	K3218	80.66666666666667	1
8	Лебедев	K3218	58.14285714285714	0
9	Леонов	K3218	75.00000000000000	0
10	Овчинников	K3218	77.75000000000000	0
11	Овчинников	K3218	65.50000000000000	0
12	Орлов	K3218	68.25000000000000	0
13	Павлова	K3218	65.83333333333333	0
14	Павлова	K3218	80.25000000000000	0
15	Павлова	K3218	72.00000000000000	0
16	Павлова	K3218	63.80000000000000	0
17	С	K3218	70.00000000000000	1

Total rows: 87 of 87

Query complete 00:00:00.055

Query Query History

```
1 select
2     session.student.surname,
3     session.student.name,
4     session.group.group_name,
5     tmp.avg_grade,
6     tmp.scholarship_given
7 from
8     --id студента в группе, его балл и стипендия
9     (select
10         attestation.student_in_group_id,
11         avg(grade) as avg_grade,
12
13     case
14         when attestation.student_in_group_id in
15             -- список студентов, сдавших с тройками
16             -- или не сдавших: остальные получают стипендию:
```

Data Output Messages Notifications

С именами для четкости.

6. Вывести список студентов, сдавших все положенные экзамены.

студент сдал все положенные экзамены, если с одной попытки сдал его на > 60.

Рассмотрим сдачу сессии в 2 семестре группой R3114

```
select distinct
surname, name
from
session.attestation
inner join session.student_in_group
    on session.attestation.student_in_group_id =
        session.student_in_group.student_in_group_id
inner join session.student
    on session.student.student_id =
        session.student_in_group.student_id
where session.attestation.student_in_group_id in
-- найденные id -> id студентов в список
(select distinct
    student_in_group_id
from
    session.attestation
where attestation.student_in_group_id not in

-- студенты из группы, не сдавшие хотя бы
-- 1 экз с 1 раза:
(select student_in_group_id
from
    session.attestation
where student_in_group_id in
    -- student_in_group_id тех студентов,
    -- у которых нужно проверить аттестацию
    (select
        student_in_group_id
    from session.student_in_group
    where group_id in
        -- в какой учебной группе нужно проверить
сдачу
        (select group_id from session.group
        where
            semester = 2
            and
                group_name = 'R3114'))
and grade < 60 and attempt_count = 1)

and student_in_group_id in
(select
```

сдачу

```
        student_in_group_id
from session.student_in_group
where group_id in
        -- в какой учебной группе нужно проверить

        (select group_id from session.group
where
        semester = 2
and
        group_name = 'R3114')))
```

Query Query History

```
1 select distinct
2     surname, name
3 from
4     session.attestation
5     inner join session.student_in_group
6         on session.attestation.student_in_group_id =
7             session.student_in_group.student_in_group_id
8     inner join session.student
9         on session.student.student_id =
10            session.student_in_group.student_id
11 where session.attestation.student_in_group_id in
12     -- найденные id -> id студентов в список
13     (select distinct
14         student_in_group_id
15     from
16         session.attestation
```

Data Output Messages Notifications



	surname character varying (30) 🔒	name character varying (30) 🔒
1	Полякова	Анна
2	Смирнов	Илья
3	Орлов	Сергей
4	Леонов	Михаил
5	Полякова	Екатерина
6	Орлов	Илья
7	Миронова	Ангелина
8	Осипов	Дмитрий
9	Осипов	Максим
10	Жданова	Наталья

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

Рассмотрим группы, сдававшие зимнюю сессию 2018 года.

```
select
surname,
name,
session.group.group_name,
session.student_in_group.student_in_group_id,
tmp.avg_grade
from

(select
avg(grade) as avg_grade,
session.student_in_group.group_id,
session.student_in_group.student_in_group_id
from session.attestation
inner join session.student_in_group
on session.attestation.student_in_group_id =
session.student_in_group.student_in_group_id
where
attestation_date > '2018-12-31' and attestation_date < '2019-
02-01'
group by
session.student_in_group.student_in_group_id) as tmp,

(select max(avg_grade) as group_max, group_id from
(select
    avg(grade) as avg_grade,
    session.student_in_group.group_id,
    session.student_in_group.student_in_group_id
from session.attestation
inner join session.student_in_group
on session.attestation.student_in_group_id =
session.student_in_group.student_in_group_id
where
    attestation_date > '2018-12-31' and attestation_date <
'2019-02-01'
group by
    session.student_in_group.student_in_group_id
) as tmp
group by
    group_id
) as tmp2,
```

```
session.student,  
session.student_in_group,  
session.group
```

```
where
```

```
-- условие отбора студента
```

```
tmp2.group_max = tmp.avg_grade
```

```
and
```

```
-- условия соединения таблиц
```

```
session.student.student_id =
```

```
session.student_in_group.student_id
```

```
and
```

```
tmp2.group_id = session.group.group_id
```

```
and
```

```
tmp.student_in_group_id =
```

```
session.student_in_group.student_in_group_id
```

```
and
```

```
session.group.group_id = session.student_in_group.group_id
```

```
and
```

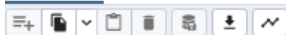
```
tmp.group_id = tmp2.group_id
```

```

1 select
2     surname,
3     name,
4     session.group.group_name,
5     session.student_in_group.student_in_group_id,
6     tmp.avg_grade
7 from
8
9 (select
10     avg(grade) as avg_grade,
11     session.student_in_group.group_id,
12     session.student_in_group.student_in_group_id
13 from session.attestation
14 inner join session.student_in_group
15     on session.attestation.student_in_group_id = session.student_in_group.student_in_group_id
16 where
17     attestation_date > '2018-12-31' and attestation_date < '2019-02-01'
18 group by
19     session.student_in_group.student_in_group_id) as tmp,
20
21 (select max(avg_grade) as group_max, group_id from
22     (select
23         avg(grade) as avg_grade,
24         session.student_in_group.group_id,
25         session.student_in_group.student_in_group_id
26     from session.attestation
27     inner join session.student_in_group
28         on session.attestation.student_in_group_id = session.student_in_group.student_in_group_id
29     where
30         attestation_date > '2018-12-31' and attestation_date < '2019-02-01'
31     group by
32         session.student_in_group.student_in_group_id
33     ) as tmp
34     group by
35         group_id
36 ) as tmp2,
37
38 session.student,
39 session.student_in_group,
40 session.group
41
42 where
43     -- условие отбора студента
44     tmp2.group_max = tmp.avg_grade
45 and
46     -- условия соединения таблиц
47     session.student.student_id = session.student_in_group.student_id
48 and
49     tmp2.group_id = session.group.group_id
50 and
51     tmp.student_in_group_id = session.student_in_group.student_in_group_id
52 and
53     session.group.group_id = session.student_in_group.group_id
54 and
55     tmp.group_id = tmp2.group_id
56

```

Data Output Messages Notifications



	surname character varying (30)	name character varying (30)	group_name character varying (10)	student_in_group_id integer	avg_grade numeric
1	Сидорова	Виктор	M3213	257	86.000000000000000000
2	Лебедева	Максим	K3218	445	94.500000000000000000
3	Жданова	Наталья	R3214	82	89.333333333333333333
4	Смирнова	София	V3220	610	88.333333333333333333

Представления

Задание 3

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

```
create view session.task1 as
select
concat(session.teacher.surname, ' ',
        substring(session.teacher.name, 1, 1), '. ',
        substring(session.teacher.patronymic, 1, 1), '.') as
teacher_name,
concat(session.student.surname, ' ',
        substring(session.student.name, 1, 1), '. ',
        substring(session.student.patronymic, 1, 1), '.') as
student_name,
attestation_date
from
session.teacher,
session.student,
session.student_in_group,
(select teacher_id, student_in_group_id, attestation_date
from session.attestation
where
student_in_group_id in
(select distinct session.attestation.student_in_group_id from
session.attestation
inner join session.student_in_group
on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
where
grade < 60
and
attempt_count = 1
and
attestation_date > '2017-12-31'
and
attestation_date < '2018-02-01'
)
and
attempt_count = 2
) as list

where
session.teacher.teacher_id = list.teacher_id
and
```

```

session.student_in_group.student_in_group_id =
list.student_in_group_id
and
session.student.student_id =
session.student_in_group.student_id
order by
teacher_name, student_name

```

Query		Query History	
1		select * from session.task1	
Data Output		Messages	
		Notifications	
	teacher_name text	student_name text	attestation_date date
1	Алексеев Д. В.	Виноградов Г. В.	2018-01-30
2	Алексеев Д. В.	Котов О. И.	2018-01-30
3	Алексеев Д. В.	Малышев А. О.	2018-01-30
4	Алексеев Д. В.	Орлов С. О.	2018-01-30
5	Алексеев Д. В.	Павлова А. В.	2018-01-30
6	Алексеев Д. В.	Павлова П. В.	2018-01-30
7	Алексеев Д. В.	Сидоров А. О.	2018-01-30
8	Волкова С. А.	Котов О. И.	2018-01-29
9	Волкова С. А.	Котов С. М.	2018-01-29
10	Волкова С. А.	Леонов К. Ю.	2018-01-29
11	Волкова С. А.	Малышев К. О.	2018-01-29
12	Волкова С. А.	Сидоров И. Б.	2018-01-29
Total rows: 58 of 58		Query complete 00:00:00.122	

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

```
create view session.task2 as
select distinct
session.student.student_id,
concat(surname, ' ',
       substring(name, 1, 1), '. ',
       substring(patronymic, 1, 1), '.') as student_name
from session.attestation
inner join session.student_in_group
on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
inner join session.student
on session.student.student_id =
session.student_in_group.student_id
where
attempt_count = 3
and
grade < 60
```

Query

Query History

1

select * from session.task2

Data Output

Messages

Notifications

	student_id integer	student_name text
1	1	Павлова А. В.
2	26	Осипов М. И.
3	36	Жданова А. В.
4	56	Алексеев Д. И.
5	71	Овчинников Д. С.
6	72	Лебедев М. М.

INSERT

Добавить по 1 лаборатории вместимостью в 40 человек в корпуса, где лабораторных классов меньше 5, и присвоить им специальный номер 5000.

Query

Query History

↗

Data Output

≡

📄

▼

📋

🗑️

🔄

📶

📈

	classroom_id [PK] integer	capacity integer	room integer	building_id integer	classroom_type session.classroom_type
1	4	11	2257	1	Лаборатория
2	70	89	2234	1	Лаборатория
3	57	56	5479	1	Лаборатория
4	82	40	4281	1	Лаборатория
5	73	68	2292	1	Лаборатория
6	100	39	1244	2	Лаборатория
7	9	92	5349	2	Лаборатория
8	24	56	5469	2	Лаборатория
9	30	29	6268	2	Лаборатория
10	47	8	3148	2	Лаборатория
11	71	20	3163	2	Лаборатория
12	35	32	6321	3	Лаборатория
13	59	13	3282	3	Лаборатория
14	25	32	6330	3	Лаборатория
15	37	17	1158	3	Лаборатория
16	54	15	4196	4	Лаборатория
17	10	31	4186	4	Лаборатория
18	20	53	3426	4	Лаборатория
19	75	70	4248	4	Лаборатория
20	45	99	3469	4	Лаборатория
21	92	19	6288	4	Лаборатория
22	32	27	5493	5	Лаборатория
23	76	13	4347	5	Лаборатория
24	61	35	3182	5	Лаборатория
25	62	81	6177	5	Лаборатория
26	12	73	6414	5	Лаборатория
27	60	100	1242	5	Лаборатория
28	74	90	2483	6	Лаборатория
29	26	49	4148	6	Лаборатория

Messages

Notifications

Total rows: 29 of 29

Query complete 00:00:00.047

Запрос:

Query Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

```
insert into session.classroom
(capacity, room, building_id, classroom_type)

select
    40, 5000, tmp.building_id, 'Лаборатория'
from
    (select distinct building_id
     from session.classroom
     where
         classroom_type = 'Лаборатория'
     group by building_id
     having count(*) < 5) as tmp
```

Messages Notifications

INSERT 0 2

Query returned successfully in 57 msec.

```
insert into session.classroom
(capacity, room, building_id, classroom_type)

select
    40, 5000, tmp.building_id, 'Лаборатория'
from
    (select distinct building_id
     from session.classroom
     where
         classroom_type = 'Лаборатория'
     group by building_id
     having count(*) < 5) as tmp
```

Результат:

Query		Query History		Data Output	
1 select * from session.classroom					
2 where classroom_type = 'Лаборатория'					
3 order by building_id					
	classroom_id [PK] integer	capacity integer	room integer	building_id integer	classroom_type session.classroom_type
2	57	56	5479	1	Лаборатория
3	70	89	2234	1	Лаборатория
4	73	68	2292	1	Лаборатория
5	82	40	4281	1	Лаборатория
6	100	39	1244	2	Лаборатория
7	9	92	5349	2	Лаборатория
8	47	8	3148	2	Лаборатория
9	24	56	5469	2	Лаборатория
10	30	29	6268	2	Лаборатория
11	71	20	3163	2	Лаборатория
12	35	32	6321	3	Лаборатория
13	101	40	5000	3	Лаборатория
14	59	13	3282	3	Лаборатория
15	37	17	1158	3	Лаборатория
16	25	32	6330	3	Лаборатория
17	20	53	3426	4	Лаборатория
18	45	99	3469	4	Лаборатория
19	54	15	4196	4	Лаборатория
20	92	19	6288	4	Лаборатория
21	75	70	4248	4	Лаборатория
22	10	31	4186	4	Лаборатория
23	60	100	1242	5	Лаборатория
24	32	27	5493	5	Лаборатория
25	62	81	6177	5	Лаборатория
26	61	35	3182	5	Лаборатория
27	12	73	6414	5	Лаборатория
28	76	13	4347	5	Лаборатория
29	74	90	2483	6	Лаборатория
30	102	40	5000	6	Лаборатория
31	26	49	4148	6	Лаборатория
Total rows: 31 of 31		Query complete 00:00:00.046			

Теперь суммарно на 2 лаборатории больше (по одной новой в 3 и 6 корпусах).

UPDATE

Для студента Сидорова Виктора Игоревича поменять оценку за предмет «Социология» (новое значение – 75 баллов), полученную во время зимней сессии 2018 года при 2-ой попытке сдачи экзамена.

```
update session.attestation
set grade = 75
where
student_in_group_id =
(select session.attestation.student_in_group_id
from session.attestation
inner join session.student_in_group
    on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
inner join session.student
    on session.student.student_id =
session.student_in_group.student_id
inner join session.edu_plan_subject
    on session.attestation.edu_plan_subject_id =
session.edu_plan_subject.edu_plan_subject_id
inner join session.subject
    on session.subject.subject_id =
session.edu_plan_subject.subject_id
where
    surname = 'Сидоров'
    and
    name = 'Виктор'
    and
    patronymic = 'Игоревич'
    and
    attestation_date > '2017-12-31'
    and
    attestation_date < '2018-02-01'
    and attempt_count = 2
    and
    subject_name = 'Социология')
```

Query

Query History

```

1 update session.attestation
2 set grade = 75
3 where
4 student_in_group_id =
5     (select session.attestation.student_in_group_id
6      from session.attestation
7      inner join session.student_in_group
8          on session.student_in_group.student_in_group_id = session.attestation.student_in_group_id
9      inner join session.student
10         on session.student.student_id = session.student_in_group.student_id
11      inner join session.edu_plan_subject
12         on session.attestation.edu_plan_subject_id = session.edu_plan_subject.edu_plan_subject_id
13      inner join session.subject
14         on session.subject.subject_id = session.edu_plan_subject.subject_id
15      where
16          surname = 'Сидоров'
17          and
18          name = 'Виктор'
19          and

```

Data Output

Messages

Notifications

UPDATE 4

Query returned successfully in 35 msec.

Выполнение запроса на проверку оценки студента:

Query

Query History

```

1 select surname, name, patronymic, grade, attempt_count, subject_name
2 from session.attestation
3 inner join session.student_in_group
4     on session.student_in_group.student_in_group_id = session.attestation.student_in_group_id
5 inner join session.student
6     on session.student.student_id = session.student_in_group.student_id
7 inner join session.edu_plan_subject
8     on session.attestation.edu_plan_subject_id = session.edu_plan_subject.edu_plan_subject_id
9 inner join session.subject
10    on session.subject.subject_id = session.edu_plan_subject.subject_id
11 where
12     surname = 'Сидоров'
13     and
14     name = 'Виктор'
15     and
16     patronymic = 'Игоревич'
17     and
18     attestation_date > '2017-12-31'
19     and
20     attestation_date < '2018-02-01'
21     and attempt_count = 2
22     and
23     subject_name = 'Социология'
24 --

```

Data Output

Messages

Notifications

≡+

📄

⌵

📋

🗑️

🔍

⬇️

📶

	<div>surname</div> <div>character varying (30) 🔒</div>	<div>name</div> <div>character varying (30) 🔒</div>	<div>patronymic</div> <div>character varying (30) 🔒</div>	<div>grade</div> <div>integer 🔒</div>	<div>attempt_count</div> <div>integer 🔒</div>	<div>subject_name</div> <div>character varying (30) 🔒</div>
1	Сидоров	Виктор	Игоревич	75	2	Социология


```
select surname, name, patronymic, grade, attempt_count,
subject_name
from session.attestation
inner join session.student_in_group
    on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
inner join session.student
    on session.student.student_id =
session.student_in_group.student_id
inner join session.edu_plan_subject
    on session.attestation.edu_plan_subject_id =
session.edu_plan_subject.edu_plan_subject_id
inner join session.subject
    on session.subject.subject_id =
session.edu_plan_subject.subject_id
where
    surname = 'Сидоров'
    and
    name = 'Виктор'
    and
    patronymic = 'Игоревич'
    and
    attestation_date > '2017-12-31'
    and
    attestation_date < '2018-02-01'
    and attempt_count = 2
    and
    subject_name = 'Социология'
```

DELETE

Для студента Сидорова Виктора Игоревича удалить запись о 3 попытке сдачи предмета «Социология» в связи с успешной аттестацией во время 2-ой попытки.

Query	Query History
<pre>1 delete from session.attestation 2 where student_in_group_id = 3 (select session.attestation.student_in_group_id 4 from session.attestation 5 inner join session.student_in_group 6 on session.student_in_group.student_in_group_id = session.attestation.student_in_group_id 7 inner join session.student 8 on session.student.student_id = session.student_in_group.student_id 9 inner join session.edu_plan_subject 10 on session.attestation.edu_plan_subject_id = session.edu_plan_subject.edu_plan_subject_id 11 inner join session.subject 12 on session.subject.subject_id = session.edu_plan_subject.subject_id 13 where 14 surname = 'Сидоров' 15 and 16 name = 'Виктор' 17 and 18 patronymic = 'Игоревич' 19 and 20 attestation_date > '2017-12-31' 21 and 22 attestation_date < '2018-02-01' 23 and attempt_count = 3 24 and 25 subject_name = 'Социология') 26 and attempt_count = 3 27</pre>	
Data Output	Messages
	<pre>DELETE 1 Query returned successfully in 119 msec.</pre>

```
delete from session.attestation
where student_in_group_id =
(select session.attestation.student_in_group_id
from session.attestation
inner join session.student_in_group
    on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
inner join session.student
    on session.student.student_id =
session.student_in_group.student_id
inner join session.edu_plan_subject
    on session.attestation.edu_plan_subject_id =
session.edu_plan_subject.edu_plan_subject_id
inner join session.subject
```

```

        on session.subject.subject_id =
session.edu_plan_subject.subject_id
where
    surname = 'Сидоров'
    and
    name = 'Виктор'
    and
    patronymic = 'Игоревич'
    and
    attestation_date > '2017-12-31'
    and
    attestation_date < '2018-02-01'
    and attempt_count = 3
    and
    subject_name = 'Социология')
and attempt_count = 3

```

Проверка:

Query	Query History
<pre> 1 select session.attestation.student_in_group_id 2 from session.attestation 3 inner join session.student_in_group 4 on session.student_in_group.student_in_group_id = session.attestation.student_in_group_id 5 inner join session.student 6 on session.student.student_id = session.student_in_group.student_id 7 inner join session.edu_plan_subject 8 on session.attestation.edu_plan_subject_id = session.edu_plan_subject.edu_plan_subject_id 9 inner join session.subject 10 on session.subject.subject_id = session.edu_plan_subject.subject_id 11 where 12 surname = 'Сидоров' 13 and 14 name = 'Виктор' 15 and 16 patronymic = 'Игоревич' 17 and 18 attestation_date > '2017-12-31' 19 and 20 attestation_date < '2018-02-01' 21 and attempt_count = 3 22 and 23 subject_name = 'Социология' 24 and attempt_count = 3 25 26 </pre>	

Data Output	Messages	Notifications
<div> <div> <div>+</div> <div>student_in_group_id</div> <div>integer</div> </div> <div> <div>+</div> <div>student_in_group_id</div> <div>integer</div> </div> </div>		

```

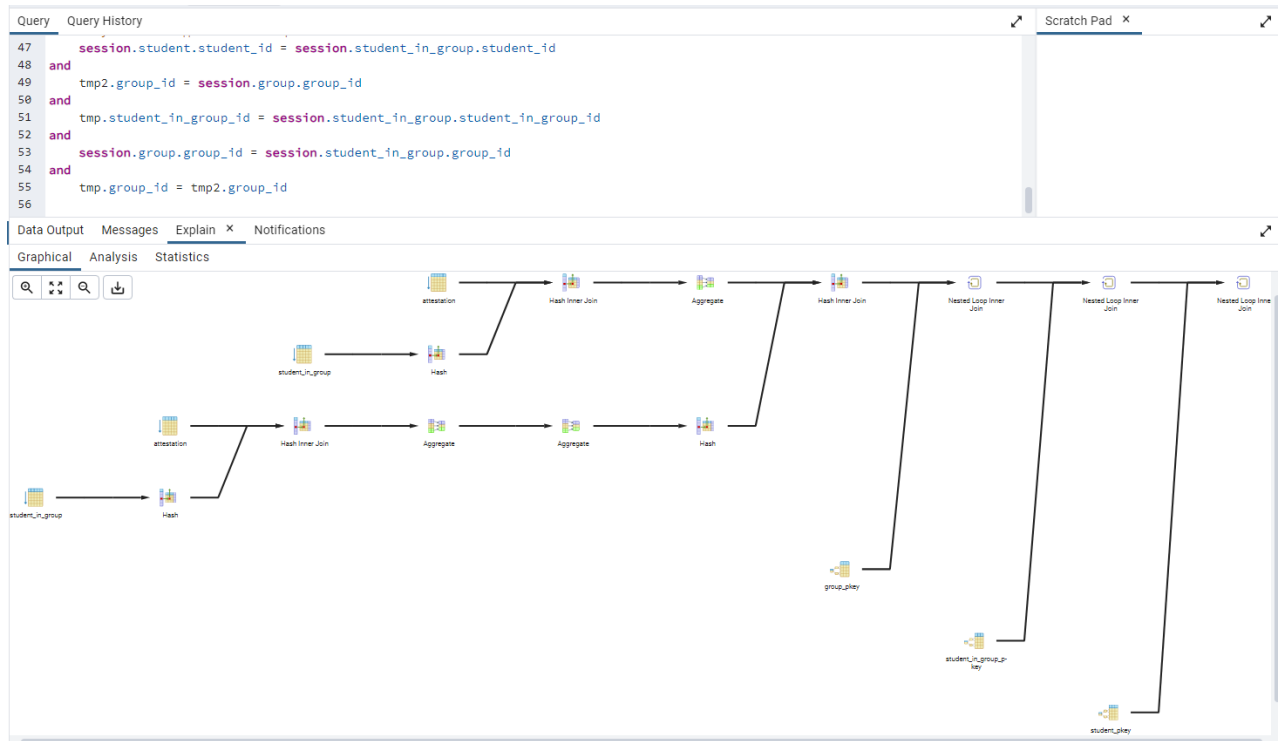
select session.attestation.student_in_group_id
from session.attestation

```

```
inner join session.student_in_group
    on session.student_in_group.student_in_group_id =
session.attestation.student_in_group_id
inner join session.student
    on session.student.student_id =
session.student_in_group.student_id
inner join session.edu_plan_subject
    on session.attestation.edu_plan_subject_id =
session.edu_plan_subject.edu_plan_subject_id
inner join session.subject
    on session.subject.subject_id =
session.edu_plan_subject.subject_id
where
    surname = 'Сидоров'
    and
    name = 'Виктор'
    and
    patronymic = 'Игоревич'
    and
    attestation_date > '2017-12-31'
    and
    attestation_date < '2018-02-01'
    and attempt_count = 3
    and
    subject_name = 'Социология'
and attempt_count = 3
```

Графическое представление запросов и история запросов

Пример использования EXPLAIN представлен с помощью запроса 7 из задания 2:



История запросов:

Query Query History

Show queries generated internally by pgAdmin? ☒ Remove Remove All

Today - 22.09.2023

select surname, name, session.group.group_name, session.studen...

13:48:39

▶ select surname, name, session.group.group_name, session.studen...

13:47:51

Yesterday - 21.09.2023

select * from session.classroom where classroom_type = 'Лабора...

20:27:55

▶ insert into session.classroom (capacity, room, building_id, cl...

20:27:43

select * from session.classroom where classroom_type = 'Лабора...

20:24:32

▶ select * from session.classroom where room_type = 'Лаборатория...

20:24:23

insert into session.classroom (capacity, room, building_id, cl...

20:22:09

insert into session.classroom (capacity, room, building_id, cl...

20:21:59

select 40, 5000, building_id, classroom_type from session.clas...

20:21:38

select 40, 5000, building_id, 'Лаборатория' from session.clas...

20:20:47

insert into session.classroom (capacity, room, building_id, cl...

20:19:57

select count(*) from session.classroom where classroom_type = ...

20:19:31

select count(*) from session.classroom where classroom_type = ...

20:19:31

22.09.2023 13:48:39

1

73 msec

Date Rows affected Duration

Copy Copy to Query Editor

```
select
  surname,
  name,
  session.group.group_name,
  session.student_in_group.student_in_group_id,
  tmp.avg_grade
from
  (select
    avg(grade) as avg_grade,
    session.student_in_group.group_id,
    session.student_in_group.student_in_group_id
  from session.attestation
  inner join session.student_in_group
    on session.attestation.student_in_group_id = session.student_in_group.student_in_group_id
  where
    attestation_date > '2018-12-31' and attestation_date < '2019-01-01'
  group by
    session.student_in_group.student_in_group_id) as tmp,
  (select max(avg_grade) as group_max, group_id from
    (select
      avg(grade) as avg_grade,
      session.student_in_group.group_id,
      session.student_in_group.student_in_group_id
    from session.attestation
    group by
      session.student_in_group.group_id,
      session.student_in_group.student_in_group_id) as tmp) as group_max
```

Индексы

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.

Простой индекс

Составим простой индекс для запроса 6: «Вывести список студентов, сдавших все положенные экзамены»

Студент сдал все положенные экзамены, если с одной попытки сдал его на > 60.

Рассмотрим сдачу сессии в 2 семестре группой R3114

```
select distinct
surname, name
from
session.attestation
inner join session.student_in_group
    on session.attestation.student_in_group_id =
        session.student_in_group.student_in_group_id
inner join session.student
    on session.student.student_id =
        session.student_in_group.student_id
where session.attestation.student_in_group_id in
-- найденные id -> id студентов в список
(select distinct
    student_in_group_id
from
    session.attestation
where attestation.student_in_group_id not in

-- студенты из группы, не сдавшие хотя бы
-- 1 экз с 1 паза:
(select student_in_group_id
from
    session.attestation
where student_in_group_id in
    -- student_in_group_id тех студентов,
    -- у которых нужно проверить аттестацию
    (select
        student_in_group_id
    from session.student_in_group
    where group_id in
        -- в какой учебной группе нужно проверить
сдачу
        (select group_id from session.group
```

```

        where
            semester = 2
        and
            group_name = 'R3114'))
and grade < 60 and attempt_count = 1)

and student_in_group_id in
(select
    student_in_group_id
from session.student_in_group
where group_id in
    -- в какой учебной группе нужно проверить
сдачу
    (select group_id from session.group
where
    semester = 2
and
    group_name = 'R3114'))))

```

Результат выполнения запроса:

Total rows: 10 of 10	Query complete 00:00:00.053
----------------------	-----------------------------

Создание индекса:

Query	Query History
1	create index attestation_index
2	on session.attestation (student_in_group_id)

Data Output	Messages	Notifications
CREATE INDEX		
Query returned successfully in 41 msec.		

Выполнение запроса с индексом:

Total rows: 10 of 10	Query complete 00:00:00.045
----------------------	-----------------------------

В файлах `explain_plan_6_without_index.svg` и `explain_plan_6_with_index.svg`, открывающихся в браузере, можно заметить индексирование по колонке в таблице `attestation`, из-за чего схемы выполнения запросов заметно отличаются.



Составной индекс

Составим составной индекс для запроса 3: «Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине».

```
select distinct session.student.surname from
session.attestation
  inner join session.student_in_group
    on session.attestation.student_in_group_id
       = session.student_in_group.student_in_group_id
  inner join session.student
    on session.student_in_group.student_id
       = session.student.student_id
where
session.attestation.grade >
  (select avg(grade) from session.attestation
  where
    -- убираем должников
    attempt_count = 1 and grade >= 60
    and edu_plan_subject_id = 4)
and edu_plan_subject_id = 4
and attempt_count = 1
```

Запрос без индексации:

Total rows: 7 of 7	Query complete 00:00:00.153
--------------------	-----------------------------

Создание индекса:

Query	Query History
1	<code>create index exam_index</code>
2	<code>on session.attestation</code>
3	<code>(attempt_count, edu_plan_subject_id, grade)</code>

Data Output	Messages	Explain	×	Notifications
CREATE INDEX				
Query returned successfully in 62 msec.				

Запрос после индексации:

Total rows: 7 of 7	Query complete 00:00:00.054
--------------------	-----------------------------

Графическая интерпретация запроса с индексом и без представлена в файлах **explain_plan_3_with_index.svg**, **explain_plan_3_without_index.svg** (открываются в браузере). В них, аналогично первому примеру с простым индексом, показана работа индекса B-tree (двоичное дерево), который используется при создании индекса по умолчанию.

Выводы

В данной лабораторной работе были рассмотрены возможности Query Tool для выполнения запросов к созданной ранее базе данных с помощью оператора SELECT, а также рассмотрены возможности изменения данных посредством операторов INSERT, UPDATE, DELETE.

Также было проведено ознакомление с представлениями, индексацией запросов и оператором EXPLAIN для выяснения деталей выполнения запросов и графического представления выполнения запросов.