

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий

Лабораторная работа №2
«Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»
по дисциплине «Проектирование и
реализация баз данных»

Выполнил:
студент II курса ИКТ
группы К3241
Хайрнасов А.К.

Проверил:
Говорова М.М.

Санкт-Петербург
2022

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.

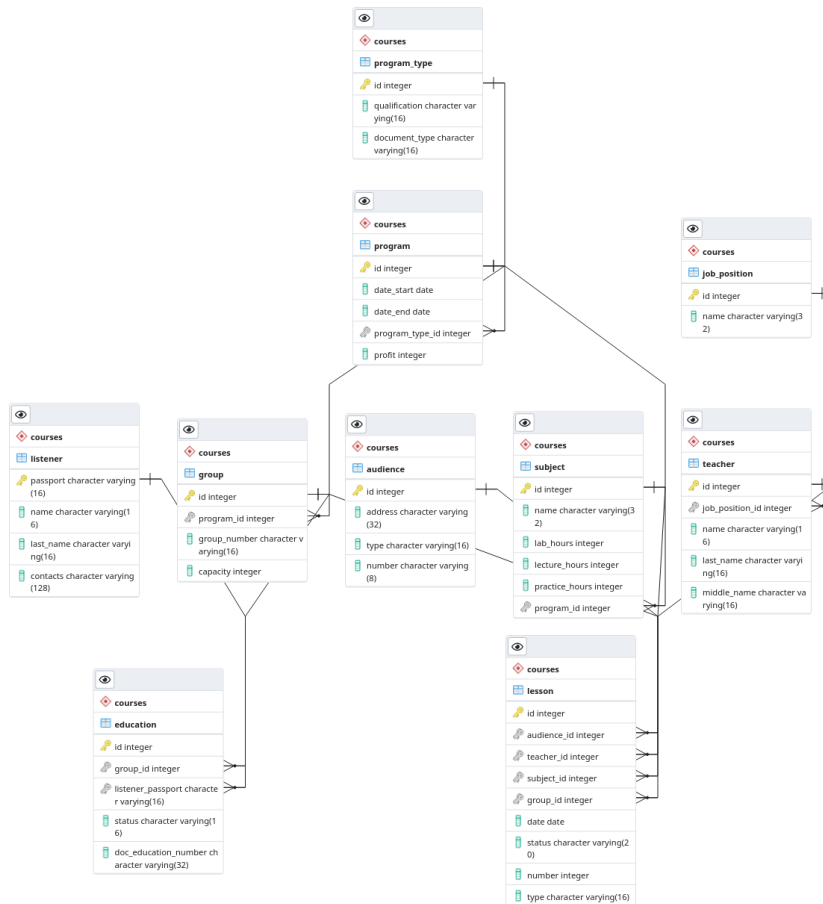
Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

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.

Выполнение:

1. Схема логической модели базы данных



2. Запросы и представления на выборку данных к базе данных

Запросы

1 - Вывести все номера групп и программы, где количество слушателей меньше 10.

```

SELECT group_number, program_id
FROM courses."group"
WHERE (SELECT COUNT(*) FROM courses.education WHERE education.group_id = courses."group".id) < 10

```

Services

Output test_db.courses.group x

db@62.84.118.19

console 417 ms

audience 977 ms

group_number	program_id
1 K3166	1

2 - Вывести список преподавателей с указанием количества программ, где они преподавали за истекший год.

```

SELECT *,
(
    SELECT COUNT(*)
    FROM (
        SELECT (
            SELECT DISTINCT (SELECT courses.program.id
                            FROM courses.program
                            WHERE courses.program.id = courses."group".program_id)
            FROM courses."group"
            WHERE courses."group".id = group_id
        )
        FROM courses.lesson
        WHERE lesson.date > CURRENT_DATE - INTERVAL '1 year' AND lesson.teacher_id = courses.teacher.id
    ) AS program_id_list
) AS programs_count
FROM courses.teacher;

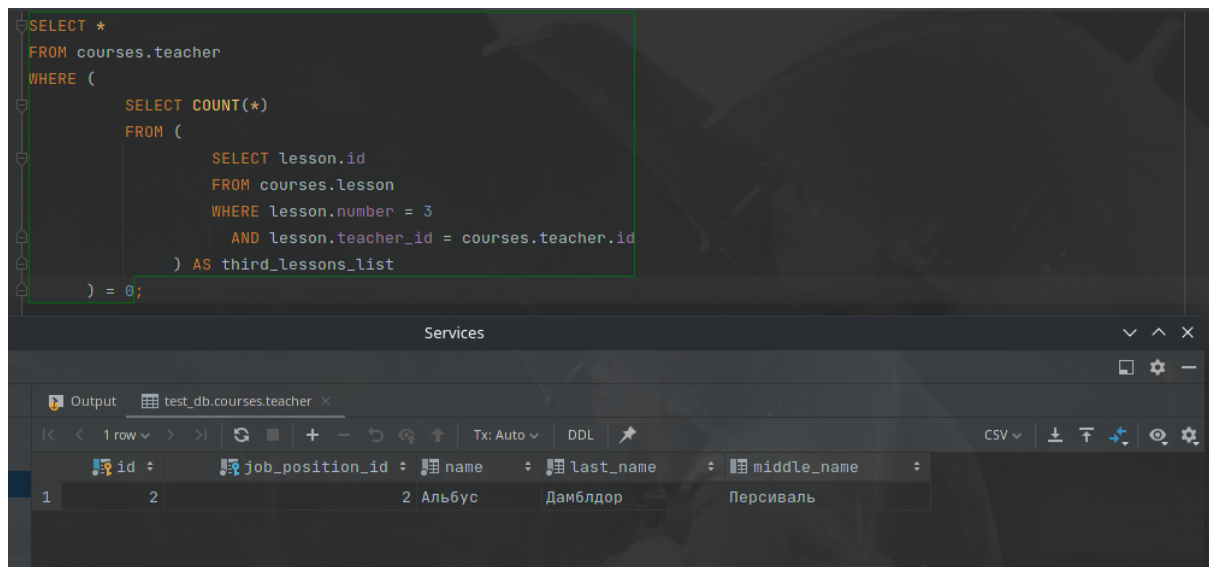
```

Services

Output Result 16 x

	id	job_position_id	name	last_name	middle_name	programs_count
1	1	2	Жмышенко	Валерий	Альбертович	1
2	2	2	Альбус	Дамблдор	Персиваль	0

3 - Вывести список преподавателей, которые не проводят занятия на третьей паре ни в один из дней недели.



4 - Вывести список свободных лекционных аудиторий на ближайший понедельник.

```
SELECT *
FROM courses.audience
WHERE (
    SELECT COUNT(*)
    FROM (
        SELECT lesson.id
        FROM courses.lesson
        WHERE lesson.audience_id = courses.audience.id
        AND EXTRACT(DOW FROM lesson.date) = 1
        AND lesson.date > CURRENT_DATE
        AND lesson.date <= CURRENT_DATE + 7
    ) AS third_lessons_list
) = 0;
```

Services

Output test_db.courses.audience x CURRENT_DATE + 7:date x

9 rows

	id	address	type	number
1	2	Пушкина, Калатушкина	Лекционная	123
2	7	Пушкина, Калатушкина	Лекционная	12
3	8	Пушкина, Калатушкина	Лекционная	54
4	5	Пушкина, Калатушкина	Лекционная	523
5	6	Пушкина, Калатушкина	Лекционная	643
6	9	Пушкина, Калатушкина	Лекционная	77
7	10	Пушкина, Калатушкина	Лекционная	32
8	3	Пушкина, Калатушкина	Лекционная	23
9	4	Пушкина, Калатушкина	Лекционная	421

5 - Вычислить общее количество обучающихся по каждой программе за последний год.

```
SELECT program.id,
(
    SELECT COUNT(*)
    FROM (
        SELECT (
            SELECT id
            FROM courses.education
            WHERE education.group_id = "group".id
        )
        FROM courses."group"
        WHERE "group".id = program.id
    ) as student_list
)
FROM courses.program
```

Services

Output Result 28 x

2 rows

	id	count
1	1	1
2	2	0

6 - Вычислить среднюю загруженность компьютерных классов в неделю за последний месяц (в часах).

```
SELECT id, (  
    SELECT COUNT(*)  
    FROM (  
        SELECT *  
        FROM courses.lesson  
        WHERE lesson.date <= CURRENT_DATE  
        AND lesson.date > CURRENT_DATE - 28  
    ) as lessons_id_list  
    ) / 4.0 AS average_classes  
FROM courses.audience  
WHERE audience.type = 'Компьютерная'
```

Services

Output Result 32 x

1 row

	id	average_binding
1	4	1.25

7 - Найти самые популярные программы за последние 3 года.


```

1 ✓ SELECT id, (
2     SELECT COUNT(*)
3     FROM(
4         SELECT (
5             SELECT id
6             FROM courses.education
7             WHERE education.group_id = "group".id
8         )
9         FROM courses."group"
10        WHERE "group".program_id = program.id
11        ) AS education_list
12    ) as education_count
13 FROM courses.program
14 WHERE program.date_start >= CURRENT_DATE - INTERVAL '3 year'
15 ORDER BY education_count DESC;

```

Services

Output Result 35 x

2 rows v

	id	education_count
1	1	1
2	2	0

Представления

1 - Для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов

```

CREATE OR REPLACE VIEW programs_data_for_abit
AS
SELECT pt.qualification, p.id, (s.lab_hours + s.lecture_hours + s.practice_hours) AS study_hours
FROM program p
JOIN subject s on p.id = s.program_id
JOIN program_type pt on pt.id = p.program_type_id

```

WHERE ORDER BY

	qualification	id	study_hours
1	Полное высшее	1	4
2	ПИ	2	39

2 - общий доход по каждой программе за последний год.

```
CREATE OR REPLACE VIEW programs_summary_profit
AS
SELECT pt.qualification,
       p.id,
       profit * (SELECT COUNT(*)
                 FROM education e
                 WHERE e.group_id = g.id) as profit
FROM program p
      JOIN "group" g on p.id = g.program_id
      JOIN program_type pt on pt.id = p.program_type_id
```

	qualification	id	profit
1	Полное высшее	1	10000
2	ПИ	2	0

3. Запросы на модификацию данных

```
1 ✓ SELECT *
2   FROM program;
3 ✓ INSERT INTO program (date_start, date_end, program_type_id, profit)
4   VALUES (DATE '2022-09-01', DATE '2023-09-01', 3, (SELECT SUM(profit) FROM program WHERE program_type_id IN (1, 2)));
5 ✓ SELECT *
6   FROM program;
7 ✓ SELECT *
8   FROM "group";
9 ✓ UPDATE "group" g
10  SET program_id = (SELECT p.id FROM program p WHERE p.program_type_id = 3)
11  WHERE (SELECT id: COUNT(*) FROM education e where e.group_id = g.id) < 3;
12 ✓ SELECT *
13   FROM "group";
14 ✓ SELECT *
15   FROM program;
16 ✓ DELETE
17   FROM program p
18   WHERE (SELECT id: COUNT(*) FROM "group" g WHERE g.program_id = p.id) = 0;
19 ✓ SELECT *
20   FROM program;
```

1 - Добавить мегапрограмму с суммарной стоимостью двух других
До

	id	date_start	date_end	program_type_id	profit
1	2	2022-09-01	2023-09-01	2	15000
2	1	2022-09-01	2023-09-01	1	10000

После

	id	date_start	date	program_type_id	profit
1	2	2022-09-01	2023-09-01	2	15000
2	1	2022-09-01	2023-09-01	1	10000
3	7	2022-09-01	2023-09-01	3	25000

2 - Переназначить группы, где менее 3 людей на новую программу

До

	id	program_id	group_number	capacity
1	1	1	K3166	40
2	2	2	110101	30

После

	id	program_id	group_number	capacity
1	1	7	K3166	40
2	2	7	110101	30

3 - Удалить программы, где 0 людей

До

	id	date_start	date_end	program_type_id	profit
1	2	2022-09-01	2023-09-01	2	15000
2	1	2022-09-01	2023-09-01	1	10000
3	7	2022-09-01	2023-09-01	3	25000

После

	id	date_start	date_end	program_type_id	profit
1	7	2022-09-01	2023-09-01	3	25000

4. Индексы и EXPLAIN

1. Запросы без индексов

Запрос №1

```
1 SELECT group_number, program_id
2 FROM courses."group"
3 WHERE (SELECT COUNT(*) FROM courses.education WHERE education.group_id = courses."group".id) < 10
4
```

Query Output Explain Messages Notifications

Graphical Analysis Statistics

```
graph LR
    subgraph SubPlan_1 [SubPlan 1]
        direction LR
        A[education_sk_2] --> B[education]
        B --> C[Aggregate]
    end
    C --> D[group]
```

Время выполнения запроса

Successfully run. Total query runtime: 147 msec.
2 rows affected.

Запрос №5

Query Editor Query History

```
1 SELECT program.id,  
2     (  
3         SELECT COUNT(*)  
4         FROM (  
5             SELECT (  
6                 SELECT id  
7                 FROM courses.education  
8                 WHERE education.group_id = "group".id  
9             )  
10            FROM courses."group"  
11            WHERE "group".id = program.id  
12        ) as student_list  
13    )  
14 FROM courses.program
```

Data Output Explain Messages Notifications

Graphical Analysis Statistics

```
graph LR
    subgraph SubPlan_1 [SubPlan 1]
        group[group] --> Aggregate[Aggregate]
    end
    Aggregate --> program[program]
```

Время работы запроса

Query Editor Query History

```

1  SELECT program.id,
2      (
3          SELECT COUNT(*)
4          FROM (
5              SELECT (
6                  SELECT id
7                  FROM courses.education
8                  WHERE education.group_id = "group".id
9              )
10             FROM courses."group"
11             WHERE "group".id = program.id
12         ) as student_list
13     )
14 FROM courses.program

```

Data Output Explain Messages Notifications

Graphical Analysis Statistics

Создание простого индекса для второго запроса (запрос №4)

```
CREATE INDEX index1 ON education(id);
```

Время выполнения 1

Successfully run. Total query runtime: 111 msec.
2 rows affected.

Создание составного индекса для второго запроса (запрос №5)

```
CREATE INDEX index2 ON "group"(id);
```

Время выполнения

Successfully run. Total query runtime: 181 msec.
1 rows affected.

Удаление индексов

Удаление первого индекса

```
BEGIN;  
DROP INDEX index1;  
DROP INDEX index2;  
COMMIT;
```

Вывод: с помощью SQL запросов были показаны, изменены, удалены и добавлены данные в базу. Время выполнения запросов с использованием индексов стало дольше.