Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами»

по дисциплине «Проектирование и реализация баз данных»

Автор: Таипов Т. А.

Факультет: ИКТ

Группа: К3241

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.

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.

1. Запросы к базе данных

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

SELECT "group".id, current_members, curriculum.id AS curriculum_id, program.id AS program_id

FROM "group"

JOIN curriculum ON curriculum.id = "group".curriculum_id

JOIN program ON program.id = curriculum.program_id

Where current members < 10;

	id integer	current_members integer	curriculum_id integer	program_id integer
1	2	9	3	3
2	4	7	2	2

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

SELECT

l.id AS lecturer_id,

1.full_name AS lecturer_name,

COUNT(DISTINCT p.id) AS number_of_programs_taught

FROM

public.lecturer l

JOIN

public.schedule s ON l.id = s.lecturer_id

JOIN

public.discipline_to_curriculum dc ON s.discipline_in_curriculum_id = dc.id

JOIN

public.curriculum c ON dc.curriculum_id = c.id

JOIN

public.program p ON c.program_id = p.id

WHERE

EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT_DATE) - 1 GROUP BY

l.id, l.full name;

,	lecturer_id integer	lecturer_name character varying (50)	number_of_programs_taught bigint
1	2	lecturer2	1
2	3	lecturer3	1

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

```
SELECT
  AVG(weekly_load) / 4.41 AS average_weekly_load
FROM (
  SELECT
    SUM(90) AS weekly_load
  FROM
    public.schedule s
  WHERE
    s.date >= CURRENT_DATE - INTERVAL '1 month'
    AND s.audience_id IN (
      SELECT id
      FROM public.audience
      WHERE type = 'computer_lab'
) AS weekly_subquery;
       average_weekly_load
       numeric
1
        81.6326530612244898
```

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

```
SELECT s.id, time, lecturer.id AS lecturer_id, lecturer.full_name FROM schedule s
JOIN lecturer ON lecturer.id = s.lecturer_id
WHERE time != '12:00:00'
```

	id integer	time time without time zone	integer	full_name character varying (50)
1	1	10:00:00	1	lecturer1
2	2	11:00:00	2	lecturer2
3	4	13:00:00	4	lecturer4

• Вывести список свободных лекционных аудиторий на ближайший понедельник.

SELECT

```
a.id AS audience_id,a."number" AS audience_number
```

FROM

public.audience a

WHERE

```
a.type = 'lecture'
```

```
AND a.id NOT IN (
SELECT DISTINCT
```

s.audience_id

FROM

public.schedule s

WHERE

EXTRACT(DOW FROM s.date) = 1

AND s.date >= CURRENT_DATE

AND s.date < CURRENT_DATE + INTERVAL '7 days'



• Вычислить общее количество обучающихся по каждой программе за последний год.

SELECT p.name AS program_name, COUNT(stg.id) AS total_students

FROM public.program p

JOIN public.curriculum c ON p.id = c.program_id

JOIN public."group" g ON c.id = g.curriculum_id

JOIN public.student_to_group stg ON g.id = stg.group_id

WHERE g.date_of_creation BETWEEN CURRENT_DATE - INTERVAL '1' YEAR AND CURRENT_DATE

GROUP BY p.name

ORDER BY total_students DESC;

	program_name character varying (50)	total_students bigint
1	program1	4
2	program3	4
3	program2	1

• Найти самые популярные программы за последние 3 года.

```
WITH ProgramsWithRanks AS (
  SELECT
    p.id AS program_id,
    p.name AS program_name,
    COUNT(sg.student_id) AS total_students,
    RANK() OVER (ORDER BY COUNT(sg.student_id) DESC) AS rnk
  FROM
    public.program p
  JOIN
    public.curriculum c ON p.id = c.program_id
  JOIN
    public."group" g ON c.id = g.curriculum_id
  JOIN
    public.student_to_group sg ON g.id = sg.group_id
  WHERE
    g.date_of_creation >= (CURRENT_DATE - INTERVAL '3 years')
  GROUP BY
    p.id, p.name
)
SELECT
  program_id,
  program_name,
  total_students
FROM
```

Programs With Ranks

WHERE

rnk = 1;

	program_name character varying (50)	number_of_students bigint	â
1	program1		4
2	program3		4

2. Представления

• Для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов;

```
CREATE OR REPLACE VIEW public.potential_students_view AS
SELECT

p.id AS program_id,
p.name AS program_name,
d.id AS discipline_id,
d.name AS discipline_name,
d.type AS discipline_type,
d.duration AS hours_per_week
FROM
public.program p
JOIN
public.curriculum c ON p.id = c.program_id
JOIN
public.discipline_to_curriculum dc ON c.id = dc.curriculum_id
JOIN
```

pub	lic.discipline	d	ON	dc.discipline_id	=	d.id;
	program_id integer	program_name character varying (50)	discipline_id integer	discipline_name character varying (50)	discipline_type disciplinetype	hours_per_week integer
1	1	program1	1	discipline1	lecture	1
2	1	program1	2	discipline1	practice	1
3	2	program2	3	discipline2	lecture	1
4	2	program2	4	discipline2	practice	1

• общих доход по каждой программе за последний год.

CREATE OR REPLACE VIEW public.program_revenue

AS

SELECT p.id AS program_id,

p.name AS program_name,

sum(c.contract_price) AS total_revenue

FROM program p

JOIN curriculum c ON p.id = c.program_id

JOIN "group" g ON g.curriculum_id = c.id

JOIN student_to_group sg ON g.id = sg.group_id

WHERE g.date_of_creation >= (CURRENT_DATE - '1 year'::interval year) AND sg.type = 'contract'::studenttogrouptype

GROUP BY p.id, p.name;

	program_id integer	program_name character varying (50)	total_revenue bigint
1	1	program1	100000
2	2	program2	130000
3	3	program3	450000

3. Запросы на модификацию (с подзапросами)

• Добавить нового студента в группу

INSERT INTO public.student_to_group (student_id, group_id, start_date, end_date, status)
VALUES (
 (SELECT id FROM public.student WHERE full_name = 'Pupin Vasilis Vasiliev'),
 (SELECT id FROM public."group" WHERE name = 'group4'),
 '2023-01-01',
 NULL,
 'studying');

	id integer	student_id integer	group_id integer	start_date date	end_date date	status studenttogroupstatus
1	10	10	4	2023-01-01	[null]	studying

• Изменить дату окончания у студента в группе

UPDATE public.student_to_group SET end_date = '2023-12-31', status = 'graduated' WHERE

student_id = (SELECT id FROM public.student WHERE full_name = 'Pupin Vasilis Vasiliev') AND group_id = (SELECT id FROM public."group" WHERE name = 'group4');

	id integer	student_id integer	group_id integer	start_date date	end_date date	status studenttogroupstatus
1	10	10	4	2023-01-01	2023-12-31	graduated

• Удалить все записи о студенте из группы:

DELETE FROM public.certification
WHERE student_id = (SELECT id FROM public.student WHERE full_name = 'Petrov Petr Petrovich');
DELETE FROM public.student_to_group
WHERE student_id = (SELECT id FROM public.student WHERE full_name = 'Petrov Petr Petrovich');

DELETE 1

Query returned successfully in 76 msec.

4. Индексы

Тестовый запрос:

EXPLAIN ANALYZE

SELECT *

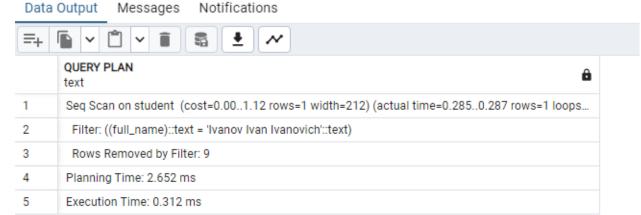
FROM public.student WHERE full_name = 'Ivanov Ivan Ivanovich';

EXPLAIN ANALYZE

SELECT * FROM public.student WHERE full_name = 'Ivanov Ivan Ivanovich' AND phone_number='+77529123456';

Без индекса:

- 1 EXPLAIN ANALYZE
- 2 SELECT * FROM public.student WHERE full_name = 'Ivanov Ivan Ivanovich';

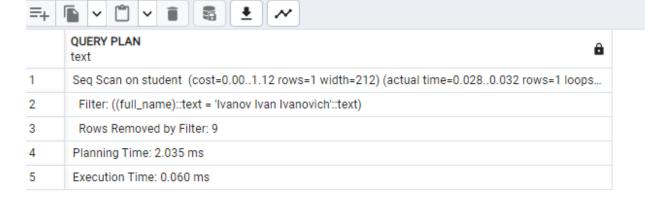


С простым индексом

CREATE INDEX idx_student_name ON public.student (full_name);:

- 1 EXPLAIN ANALYZE
- 2 SELECT * FROM public.student WHERE full_name = 'Ivanov Ivan Ivanovich';

Data Output Messages Notifications



С составным индексом

CREATE INDEX idx_student_name_phone ON public.student (full_name, phone_number);:

```
1 EXPLAIN ANALYZE
2 SELECT * FROM public.student WHERE full_name = 'Ivanov Ivan Ivanovich' A
3
Data Output Messages Notifications
      QUERY PLAN
                                                                                               ۵
      text
       Seq Scan on student (cost=0.00..1.15 rows=1 width=212) (actual time=0.015..0.018 rows=1 loops=1)
1
2
       Filter: (((full_name)::text = 'lvanov Ivan Ivanovich'::text) AND ((phone_number)::text = '+77529123456'::te...
3
       Rows Removed by Filter: 9
4
      Planning Time: 1.966 ms
5
       Execution Time: 0.031 ms
```

Вывод

В ходе лабораторной работы я освоил практические навыки по выполнению select, insert, delete и update запросов. Также, научился делать представления и индексы. В ходе анализа в первом запросе индексы позволили сократить время выполнения запроса.