

Лабораторная работа № 4

**«Запросы на выборку и модификацию данных.
Представления. Работа с индексами»**

Выполнила: Анисимова Ксения Сергеевна

Группа: К3241

Преподаватель: Говорова Марина Михайловна

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:
 1. Вывести все номера групп и программы, где количество слушателей меньше 10.
 2. Вывести список преподавателей с указанием количества программ, где они преподавали за истекший год.
 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.
 5. Вычислить общее количество обучающихся по каждой программе за последний год.
 6. Вычислить среднюю загруженность компьютерных классов в неделю за последний месяц (в часах).
 7. Найти самые популярные программы за последние 3 года.
- 2) Создать представление:
 1. для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов;
 2. общих доход по каждой программе за последний год.
- 3) Создать запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;
- 4) Создать индексы. Выполнить запросы без индекса и создать планы запросов. Выполнить создание индексов. Выполнить запросы с индексами и создать планы запросов. Сравнить время выполнения запросов. Удалить индексы.

Ход работы:

- 1) Создать запросы:
 1. Вывести все номера групп и программы, где количество слушателей меньше 10.

```
select g.group_name, p.program_name
from courses.group g
      join courses.education e on e.group_name = g.group_name
      join courses.program p on g.program_id = p.program_id
group by g.group_name, p.program_id
having count (e.listener_number) < 10 |
```

Select g.group_name, p.program_name from courses.group g join courses.education e on e.group_name = g.group_name join courses.program p on g.program_id = p.program_id group by g.group_name, p.program_id having count (e.listener_number) < 10

	group_name character varying	program_name character
1	M12	Дополнительные главы высшей математики - 1 ...
2	M21	Линейная алгебра - 2
3	M23	Линейная алгебра - 2
4	K33	Теория вероятностей и статистика ...

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

```
select t.teacher_name, count(d.program_id) as programs
from courses.teacher t
join courses.discipline d on t.teacher_number = d.teacher_number
join courses.program p on d.program_id = p.program_id
where p.start_date between '2023-01-01' and '2023-12-31' or p.end_date between '2023-01-01' and '2023-12-31'
group by t.teacher_name
```


```
select t.teacher_name, count(d.program_id) as programs from courses.teacher t join courses.discipline d on
t.teacher_number = d.teacher_number join courses.program p on d.program_id = p.program_id where
p.start_date between '2023-01-01' and '2023-12-31' or p.end_date between '2023-01-01' and '2023-12-31'
group by t.teacher_name
```

	teacher_name character varying (32)	programs bigint
1	Poco Maria Locovna	2
2	Hvosh Svetlana Sergeevna	1
3	Polya Polina Pavlovna	4

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

```
select t.teacher_name
from courses.teacher t
where t.teacher_number not in(
select distinct d.teacher_number
from courses.discipline d
join courses.timetable tt on d.discipline_id = tt.discipline_id
where tt.period = 3);
```


```
select t.teacher_name from courses.teacher t where t.teacher_number not in(select distinct d.teacher_number
from courses.discipline d join courses.timetable tt on d.discipline_id = tt.discipline_id where tt.period = 3);
```

	teacher_name character varying (32) 
1	Forge Katya Lolitovna
2	Forge Ivan Kolyntin
3	Jorjev Kolyan Kolyntin
4	Lokin Kolyan Sergeev
5	Hvosh Svetlana Sergeevna
6	Horhe Svetlana Golins
7	Shishkina Marina Gopons
8	Polya Polik Pavlovich
9	Solec Polina
10	Solec Pavel

4. Вывести список свободных лекционных аудиторий на ближайший понедельник.

```
select c.classroom_id
from courses.classroom c
where c.classroom_id not in (
    select distinct t.classroom_id
    from courses.timetable t
    where t.week_day = 'ПН' and period is not null );
```

```
select c.classroom_id from courses.classroom c where c.classroom_id not in ( select distinct t.classroom_id
from courses.timetable t where t.week_day = 'ПН' and period is not null );
```

	classroom_id [PK] integer 
1	1
2	2
3	4
4	7
5	9
6	10
7	3
8	5

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

```

select p.program_name, count(e.listener_number) as number_of_listeners
from courses.program p
    join courses.group g on p.program_id = g.program_id
    join courses.education e on g.group_name = e.group_name
where p.start_date between '2023-01-01' and '2023-12-31'
    or p.end_date between '2023-01-01' and '2023-12-31'
group by p.program_name

```

```

select p.program_name, count(e.listener_number) as number_of_listeners from courses.program p join
courses.group g on p.program_id = g.program_id join courses.education e on g.group_name = e.group_name
where p.start_date between '2023-01-01' and '2023-12-31' or p.end_date between '2023-01-01' and '2023-12-
31' group by p.program_name

```

	program_name character		number_of_listeners bigint
1	Математический анализ	...	5
2	Линейная алгебра - 1	...	11

6. Вычислить среднюю загруженность компьютерных классов в неделю (в часах).

```

select class, avg( periods*1.5) as avg_hours
from (select c.classroom_id as class, count(tt.period) as periods from courses.classroom c
join courses.timetable tt on c.classroom_id = tt.classroom_id where c.type = 'компьютерный класс'
group by c.classroom_id) group by class
select class, avg( periods*1.5) as avg_hours
from (select c.classroom_id as class, count(tt.period) as periods from courses.classroom c
join courses.timetable tt on c.classroom_id = tt.classroom_id where c.type = 'компьютерный класс'
group by c.classroom_id) group by class

```

	class integer	avg_hours numeric
1	3	7.5000000000000000
2	5	7.5000000000000000

7. Найти самые популярные программы за последние 3 года.

```

with listener_counts as (
    select p.program_name as program,
    extract(year from p.start_date) as year,
    coalesce(count(distinct e.listener_number), 0) as listener_count
    from courses.program p
        join courses.group g on p.program_id = g.program_id
        join courses.education e on g.group_name = e.group_name
    group by p.program_name, extract(year from p.start_date)
)

select year, program
from listener_counts lc1
where listener_count = (
    select coalesce(max(lc2.listener_count), 0)
    from listener_counts lc2
    where lc2.year = lc1.year and
    lc1.year between (extract(year from current_date) - 2) and (extract(year from current_date))
);

```

```

with listener_counts as (
select p.program_name as program,
extract(year from p.start_date) as year,
coalesce(count(distinct e.listener_number), 0) as listener_count
from courses.program p
join courses.group g on p.program_id = g.program_id
join courses.education e on g.group_name = e.group_name
group by p.program_name, extract(year from p.start_date)
)

select year, program
from listener_counts lc1
where listener_count = (
select coalesce(max(lc2.listener_count), 0)
from listener_counts lc2
where lc2.year = lc1.year and
lc1.year between (extract(year from current_date) - 2) and (extract(year from current_date))
);

```

	year numeric	program character
1	2023	Линейная алгебра - 1
2	2022	Линейная алгебра - 2
3	2021	Математический анализ

2) Создать представление:

1. для потенциальных слушателей, содержащее перечень специальностей(программ), изучаемых на них дисциплин и количество часов;

```

create view program_discipline_hours as
select p.program_name as program,
d.discipline_name as discipline,
(d.lab_hours+d.prac_hours+d.lec_hours) as hours
from courses.program p
join courses.discipline d on p.program_id = d.program_id;

```

```

create view program_discipline_hours as
select p.program_name as program,
d.discipline_name as discipline,
(d.lab_hours+d.prac_hours+d.lec_hours) as hours
from courses.program p
join courses.discipline d on p.program_id = d.program_id;

```

```

select * from program_discipline_hours order by program
select * from program_discipline_hours order by program

```

	program character		discipline character varying (64)	hours integer
1	Линейная алгебра - 1	...	Алгебра. Начальный уровень	96
2	Линейная алгебра - 1	...	Линейная алгебра	96
3	Линейная алгебра - 1	...	Алгебра. Продвинутого уровня	48
4	Линейная алгебра - 1	...	Математический анализ	48
5	Линейная алгебра - 1	...	Геометрия. Начальный уровень	96
6	Линейная алгебра - 1	...	Геометрия	96
7	Линейная алгебра - 1	...	Геометрия. Продвинутого уровня	48
8	Линейная алгебра - 2	...	Алгебра. Начальный уровень	53
9	Линейная алгебра - 2	...	Линейная алгебра	53
10	Линейная алгебра - 2	...	Геометрия. Начальный уровень	53
11	Линейная алгебра - 2	...	Геометрия	53

2. общий доход по каждой программе за последний год.

```
create view program_revenue_last_year as
select p.program_name as program,
       sum(p.cost) as total_revenue_last_year
from courses.program p
where extract(year from p.end_date) = extract(year from current_date) - 1
group by p.program_name;
```

```
create view program_revenue_last_year as
select p.program_name as program,
       sum(p.cost) as total_revenue_last_year
from courses.program p
where extract(year from p.end_date) = extract(year from current_date) - 1
group by p.program_name;
```

```
select * from program_revenue_last_year
```

```
select * from program_revenue_last_year
```

	program character		total_revenue_last_year numeric
1	Дополнительные главы высшей математи...		200000
2	Математический анализ	...	100000
3	Линейная алгебра - 1	...	200000

- 3) Создать запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;

INSERT - Добавить нового студента в группу с максимальным количеством студентов

```
1 select* from courses.education
```

```
2
```

```
3
```

Data Output Messages Notifications

	listener_number [PK] bigint	status character	passport bigint	group_name character varying
1	888881	обучается	222222222	M11
2	888882	обучается	222222223	M11
3	888883	обучается	2231555399	M11
4	888884	обучается	2231144399	M11
5	888885	обучается	2221144399	M23
6	888886	обучается	2221123399	M23
7	888887	обучается	2221122299	M23
8	888888	обучается	2222222299	K33
9	888810	обучается	2222222277	K33
10	888811	обучается	2222222266	K32
11	888812	обучается	2222222255	M21
12	888813	обучается	2222222244	M12
13	888889	отчислен	2222222288	K33

```
insert into courses.education (listener_number, status, group_name, passport)
values (
  (select coalesce(max(listener_number) + 1, 1) from courses.education),
  'обучается',
  (select group_name from courses.group where volume = (select max(volume) from courses.group)),
  '3333333333'
);
```

```
insert into courses.education (listener_number, status, group_name, passport)
values (
  (select coalesce(max(listener_number) + 1, 1) from courses.education),
  'обучается',
  (select group_name from courses.group where volume = (select max(volume) from courses.group)),
  '3333333333'
);
```



```

1 select * from courses.education
2
3

```

Data Output Messages Notifications

	listener_number [PK] bigint	status character	passport bigint	group_name character varying
1	888881	обучается	222222222	M11
2	888882	обучается	222222223	M11
3	888883	обучается	2231555399	M11
4	888884	обучается	2231144399	M11
5	888885	обучается	2221144399	M23
6	888886	обучается	2221123399	M23
7	888887	обучается	2221122299	M23
8	888888	обучается	2222222299	K33
9	888810	обучается	2222222277	K33
10	888811	обучается	2222222266	K32
11	888812	обучается	2222222255	M21
12	888813	обучается	2222222244	M12
13	888889	отчислен	2222222288	K33
14	888890	обучается	3333333333	K32

UPDATE - увеличить количество часов лекций для всех дисциплин в определенной программе.

```

1 select* from courses.discipline
2
3

```

Data Output Messages Notifications

	discipline_id [PK] integer	program_id integer	discipline_type_id integer	lab_hours integer	prac_hours integer	lec_hours integer	teacher_number integer	discipline_name character varying (64)
1	1	117	14	16	16	16	222240	Алгебра - 1
2	2	116	14	32	32	32	222240	Алгебра - 1
3	3	116	14	32	32	32	222240	Алгебра - 2
4	5	116	14	16	16	16	222240	Алгебра - 3
5	6	116	14	16	16	16	222240	Алгебра - 4
6	4	117	14	16	16	16	222240	Алгебра - 2
7	7	117	15	16	16	16	222222	Геометрия - 1
8	8	117	15	16	16	16	222222	Геометрия - 2
9	9	116	15	32	32	32	222223	Геометрия - 1
10	10	116	15	32	32	32	222223	Геометрия - 2
11	11	116	15	16	16	16	222228	Геометрия - 3

UPDATE courses.discipline

SET lec_hours = lec_hours + 5

WHERE program_id = (SELECT program_id FROM courses.program WHERE program_id = 117);

UPDATE courses.discipline

SET lec_hours = lec_hours + 5

WHERE program_id = (SELECT program_id FROM courses.program WHERE program_id = 117);

```

1 select * from courses.discipline
2

```

Data Output Messages Notifications

	discipline_id [PK] integer	program_id integer	discipline_type_id integer	lab_hours integer	prac_hours integer	lec_hours integer	teacher_number integer	discipline_name character varying (64)
1	2	116	14	32	32	32	222240	Алгебра - 1
2	3	116	14	32	32	32	222240	Алгебра - 2
3	5	116	14	16	16	16	222240	Алгебра - 3
4	6	116	14	16	16	16	222240	Алгебра - 4
5	9	116	15	32	32	32	222223	Геометрия - 1
6	10	116	15	32	32	32	222223	Геометрия - 2
7	11	116	15	16	16	16	222228	Геометрия - 3
8	1	117	14	16	16	21	222240	Алгебра - 1
9	4	117	14	16	16	21	222240	Алгебра - 2
10	7	117	15	16	16	21	222222	Геометрия - 1
11	8	117	15	16	16	21	222222	Геометрия - 2

DELETE - удалить всех студентов из программ, которые завершены

```

1 select * from courses.education
2

```

Data Output Messages Notifications

	listener_number [PK] bigint	status character	passport bigint	group_name character varying
1	888881	обучается	2222222222	M11
2	888882	обучается	2222222233	M11
3	888883	обучается	2231555399	M11
4	888884	обучается	2231144399	M11
5	888885	обучается	2221144399	M23
6	888886	обучается	2221123399	M23
7	888887	обучается	2221122299	M23
8	888888	обучается	2222222299	K33
9	888810	обучается	2222222277	K33
10	888811	обучается	2222222266	K32
11	888812	обучается	2222222255	M21
12	888813	обучается	2222222244	M12
13	888889	отчислен	2222222288	K33

```

delete from courses.education
where group_name in (
    select g.group_name
    from courses.group g
    join courses.program p on g.program_id = p.program_id
    where p.end_date < current_date
);

delete from courses.education

```

```

where group_name in (
  select g.group_name
  from courses.group g
  join courses.program p on g.program_id = p.program_id
  where p.end_date < current_date
);

```

```

1 select * from courses.education
2

```

Data Output Messages Notifications

	listener_number [PK] bigint	status character	passport bigint	group_name character varying
1	888881	обучается	222222222	M11
2	888882	обучается	222222233	M11
3	888883	обучается	2231555399	M11
4	888884	обучается	2231144399	M11
5	888885	обучается	2221144399	M23
6	888886	обучается	2221123399	M23
7	888887	обучается	2221122299	M23
8	888888	обучается	2222222299	K33
9	888810	обучается	2222222277	K33
10	888812	обучается	2222222255	M21
11	888813	обучается	2222222244	M12
12	888889	отчислен	2222222288	K33

- 4) Создать индексы. Выполнить запросы без индекса и создать планы запросов. Выполнить создание индексов. Выполнить запросы с индексами и создать планы запросов. Сравнить время выполнения запросов. Удалить индексы.

Простой индекс:

Без индекса:

Query Query History

```

1 EXPLAIN ANALYZE SELECT * FROM courses.education WHERE listener_number = 888888;
2

```

Data Output Messages Notifications

	QUERY PLAN text
1	Seq Scan on education (cost=0.00..1.15 rows=1 width=180) (actual time=0.018..0.020 rows=1 loops...
2	Filter: (listener_number = 888888)
3	Rows Removed by Filter: 11
4	Planning Time: 0.287 ms
5	Execution Time: 0.419 ms

Создаем индекс:

```
1 CREATE INDEX idx_education_listener_number ON courses.education(listener_number);
2
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 196 msec.

С индексом:

```
1 EXPLAIN ANALYZE SELECT * FROM courses.education WHERE listener_number = 888888;
2
```

Data Output Messages Notifications



QUERY PLAN

text



1	Seq Scan on education (cost=0.00..1.15 rows=1 width=180) (actual time=0.176..0.178 rows=1 loops=1)
2	Filter: (listener_number = 888888)
3	Rows Removed by Filter: 11
4	Planning Time: 1.859 ms
5	Execution Time: 0.196 ms

Составной индекс:

Без индекса:

```
1 EXPLAIN ANALYZE SELECT * FROM courses.teacher WHERE (teacher_number > 222225 and occupation_id = 1122);
```

Data Output Messages Notifications



QUERY PLAN

text



1	Seq Scan on teacher (cost=0.00..1.20 rows=1 width=146) (actual time=0.091..0.093 rows=2 loops=1)
2	Filter: ((teacher_number > 222225) AND (occupation_id = 1122))
3	Rows Removed by Filter: 11
4	Planning Time: 0.297 ms
5	Execution Time: 0.113 ms

Создаем индекс:

```
1 CREATE INDEX idx_teacher_number_occupation ON courses.teacher(teacher_number, occupation_id);
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 104 msec.

С индексом:

1	EXPLAIN ANALYZE SELECT * FROM courses.teacher WHERE (teacher_number > 222225 and occupation_id = 1122);
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>QUERY PLAN</div> <div>text</div> <div>🔒</div> </div>
1	Seq Scan on teacher (cost=0.00..1.20 rows=1 width=146) (actual time=0.137..0.139 rows=2 loops...
2	Filter: ((teacher_number > 222225) AND (occupation_id = 1122))
3	Rows Removed by Filter: 11
4	Planning Time: 1.134 ms
5	Execution Time: 0.157 ms

Вывод:

В ходе выполнения лабораторной работы были разработаны и выполнены запросы на выборку данных, а также были созданы представления для базы данных PostgreSQL в соответствии с поставленной индивидуальной задачей. Кроме того, мы успешно реализовали разнообразные запросы на модификацию данных. Провели создание как простых, так и составных индексов, а также проанализировали время выполнения запросов при их использовании.