

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО»**

**Факультет инфокоммуникационных технологий**

**Дисциплина:**  
«Базы данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2  
«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И  
ИНДЕКСЫ В PostgreSQL»**

**Выполнила:**  
студентка группы К32421  
Панкова Кристина  
Сергеевна

---

(подпись)

**Проверила:**  
Говорова Марина Михайловна

---

(отметка о выполнении)

---

(подпись)

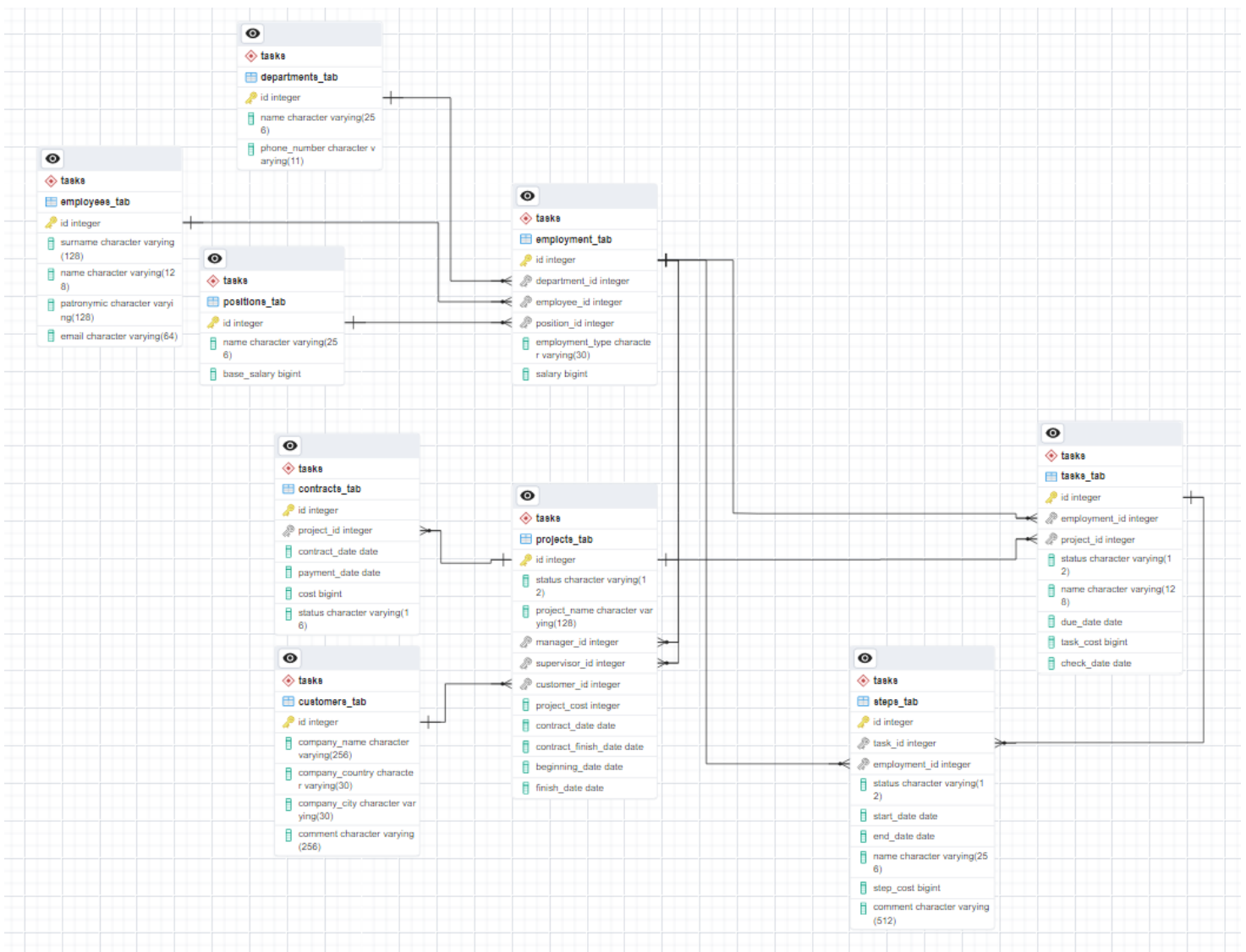
Санкт-Петербург  
2023 г.

**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 4

### Задание 2:

-- Составить список всех заданий каждого проекта с указанием организаций, отделов и исполнителей, занятых в его выполнении.

```
select proj.project_name, task.name, cust.company_name, dep.name as dep_name, empl.name||  
'||empl.surname as employee  
from tasks.tasks_tab task  
join tasks.projects_tab proj on proj.id = task.project_id  
join tasks.customers_tab cust on cust.id = proj.customer_id  
join tasks.employment_tab emp on emp.id = task.employment_id  
join tasks.employees_tab empl on empl.id = emp.employee_id  
join tasks.departments_tab dep on dep.id = emp.department_id;
```

	project_name character varying (128)	name character varying (128)	company_name character varying (256)	dep_name character varying (256)	employee text
1	project 4	task 2	Facebook	Information Technology	Aaron Gardner
2	project 4	task 3	Facebook	Legal	Brooklyn Owens
3	project 4	task 4	Facebook	Information Technology	Abigail Willis
4	project 4	task 5	Facebook	Information Technology	Natalie Hunter
5	project 4	task 6	Facebook	Product Management	Aaron Perkins
6	project 4	task 7	Facebook	Research and Development	Leo Bennett
7	project 4	task 8	Facebook	Product Management	Jasmine Bennett
8	project 4	task 9	Facebook	Human Resources	Christopher Peters
9	project5	task 2	Visa Inc.	Research and Development	David Owens
10	project5	task 3	Visa Inc.	Product Management	Madison West
11	project5	task 4	Visa Inc.	Product Management	James Mills
12	project5	task 5	Visa Inc.	Human Resources	Elena Coleman
13	project5	task 7	Visa Inc.	Research and Development	Maverick Carpenter
14	project5	task 8	Visa Inc.	Human Resources	Michael Reyes
15	project5	task 9	Visa Inc.	Information Technology	Audrey Myers
16	project 7	task 2	Toyota Moto...PetroChina	Sales	Miles Berry
17	project 7	task 3	Toyota Moto...PetroChina	Research and Development	Parker Wallace
18	project 7	task 5	Toyota Moto...PetroChina	Research and Development	Charles Holmes
19	project 7	task 8	Toyota Moto...PetroChina	Research and Development	Charles Tucker
20	project 7	task 9	Toyota Moto...PetroChina	Research and Development	Emily Owens
21	project8	task 2	JPMorgan Chase	Information Technology	Carter Perry
22	project8	task 4	JPMorgan Chase	Information Technology	Vincent Jones
Total rows: 1000 of 6626    Query complete 00:00:00.345					

1 - Результат выполнения 1 запроса

— Составить список проектов, работа над которыми была начата больше месяца назад.

```
select proj.project_name, task.name, step.start_date  
from tasks.tasks_tab task  
join tasks.projects_tab proj on proj.id = task.project_id  
join tasks.steps_tab step on step.task_id = task.id  
where coalesce(step.start_date, current_date - 35) > current_date - 31;
```

Query Query History Data Output Messages			
	project_name character varying (128)	name character varying (128)	start_date date
1	project 23	task 14	2025-12-01
2	project 23	task 14	2024-09-01
3	project 23	task 14	2025-12-04
4	project 23	task 14	2025-11-01
5	project 23	task 14	2024-09-01
6	project 23	task 17	2025-12-04
7	project 23	task 17	2024-09-06
8	project 23	task 17	2025-12-05
9	project 23	task 17	2025-11-01
10	project 23	task 17	2025-12-02
11	project 23	task 18	2025-12-08
12	project 23	task 18	2025-12-01
13	project 23	task 18	2025-11-01
14	project 23	task 18	2025-12-03
15	project 23	task 18	2024-12-06
16	project 23	task 19	2025-11-01
17	project 23	task 19	2025-12-01
18	project 23	task 19	2024-12-02
19	project 23	task 19	2025-12-04
20	project 23	task 19	2025-12-07
21	project60	task 11	2024-09-14
22	project60	task 11	2024-01-13

✓ Successfully run. Total query runtime: 635 msec. 1681 rows affected. ✕

Total rows: 1000 of 1681 Query complete 00:00:00.635 Ln 1, Col 3

## 2 - Результат выполнения 2 запроса

— Вывести список сотрудников, оклад которых превышает средний оклад сотрудников своего отдела.

```

with avg_dep as (select dep.id, avg(salary) as sal
                  from tasks.employment_tab emp
                  join tasks.employees_tab empl on empl.id = emp.employee_id
                  join tasks.departments_tab dep on dep.id = emp.department_id
                  group by dep.id
)
select empl.name||' '||empl.surname, sum(emp.salary), dep.id, avg(dep.sal)
from tasks.employment_tab emp
join tasks.employees_tab empl on empl.id = emp.employee_id
join avg_dep dep on dep.id = emp.department_id
group by empl.name||' '||empl.surname, dep.id
having sum(emp.salary) >= avg(dep.sal);

```

tasks\_track/postgres@PostgreSQL 14

Query Query History Data Output Messages

	?column? text	sum numeric	id [PK] integer	avg numeric
1	Ellie Thompson	1914860	5	1050664.188544152745
2	Roman Walker	1785267	8	1002824.390213299875
3	Tristan Perry	1116168	4	1017033.031901840491
4	Isaac Graham	1503121	4	1017033.031901840491
5	John Perry	1300941	2	1015544.658056580566
6	Alexander Banks	2109775	4	1017033.031901840491
7	Miles Alexander	1912036	1	1032209.557038834951
8	Lily Long	1225240	7	1030884.375155279503
9	Natalie Stewart	2309320	1	1032209.557038834951
10	Wyatt Willis	1663915	5	1050664.188544152745
11	Samantha George	1970928	8	1002824.390213299875
12	Audrey Lewis	1109636	1	1032209.557038834951
13	Abigail Cole	1551603	9	1043514.734449760766
14	Victoria Gray	1784642	8	1002824.390213299875
15	Mila Long	1478184	4	1017033.031901840491
16	Violet Nguyen	1152143	6	1010575.303813038130
17	Leah Watkins	1552906	1	1032209.557038834951
18	Vincent Jacobs	1968313	7	1030884.375155279503
19	Henry Anderson	2280243	7	1030884.375155279503
20	Aaliyah Roberts	1181177	3	1029088.681212121212
21	Alexander Hunter	1232533	2	1015544.658056580566
22	Antonio Carroll	1409439	4	1017033.031901840491

✓ Successfully run. Total query runtime: 391 msec. 3594 rows affected. ✕

Total rows: 1000 of 3594 Query complete 00:00:00.391 Ln 1, Col 3

3 - Результат выполнения 3 запроса

### – Найти отдел, работающий над максимальным количеством проектов.

```

with prep as (
  select dep.id, count(distinct proj.project_name) as cnt, max(count(distinct proj.project_name)) OVER ()
  as max_val
  from tasks.tasks_tab task
  join tasks.projects_tab proj on proj.id = task.project_id
  join tasks.steps_tab step on step.task_id = task.id
  join tasks.employment_tab emp on emp.id in (step.employment_id, task.employment_id,
proj.supervisor_id, proj.manager_id)
  join tasks.departments_tab dep on dep.id = emp.department_id
  group by dep.id
) select dep.name as department_name, prep.cnt as projects_count
  from prep
  join tasks.departments_tab dep on dep.id = prep.id
 where prep.cnt = prep.max_val;

```

Query Query History Data Output Messages

	department_name character varying (256)	projects_count bigint
1	Legal	1184

4 - Результат выполнения 4 запроса

**-- Составить список сотрудников, проектов, заданий, в выполнении которых они участвуют и дат предполагаемого выполнения ими заданий. Учесть сотрудников, не участвующих в проектах.**

```
select empl.name||' '||empl.surname as employee, dep.name, proj.project_name, task.name as
task_name, task.due_date
from tasks.employees_tab empl
join tasks.employment_tab emp on emp.employee_id = empl.id
join tasks.departments_tab dep on dep.id = emp.department_id
left join tasks.tasks_tab task on task.employment_id = emp.id
left join tasks.projects_tab proj on proj.id = task.project_id
left join tasks.steps_tab step on step.task_id = task.id;
```

	employee text	name character varying (256)	project_name character varying (128)	task_name character varying (128)	due_date date
381	Ezra Scott	Research and Development	project 23	task 14	2024-12-07
382	Ezra Scott	Research and Development	project 23	task 14	2024-12-07
383	Ezra Scott	Research and Development	project 23	task 14	2024-12-07
384	Ezra Scott	Research and Development	project 23	task 14	2024-12-07
385	Ezra Scott	Research and Development	project 23	task 14	2024-12-07
386	Logan Bell	Information Technology	project 23	task 17	2025-12-06
387	Logan Bell	Information Technology	project 23	task 17	2025-12-06
388	Logan Bell	Information Technology	project 23	task 17	2025-12-06
389	Logan Bell	Information Technology	project 23	task 17	2025-12-06
390	Logan Bell	Information Technology	project 23	task 17	2025-12-06
391	Andrew Ruiz	Research and Development	project 23	task 18	2025-11-01
392	Andrew Ruiz	Research and Development	project 23	task 18	2025-11-01
393	Andrew Ruiz	Research and Development	project 23	task 18	2025-11-01
394	Andrew Ruiz	Research and Development	project 23	task 18	2025-11-01
395	Andrew Ruiz	Research and Development	project 23	task 18	2025-11-01
396	Riley Nelson	Legal	project 23	task 19	2024-12-05
397	Riley Nelson	Legal	project 23	task 19	2024-12-05
398	Riley Nelson	Legal	project 23	task 19	2024-12-05
399	Riley Nelson	Legal	project 23	task 19	2024-12-05
400	Riley Nelson	Legal	project 23	task 19	2024-12-05

5 - Результат выполнения 5 запроса

**— Составить список сотрудников, не выполнивших задания в срок с указанием проектов и заданий, которые они должны были выполнить и количества дней просрочки выполнения заданий.**

```
select distinct empl.name||' '||empl.surname as employee, proj.project_name, task.name as task_name,
coalesce(step.end_date, task.due_date +3) - task.due_date as exp_days
from tasks.employees_tab empl
join tasks.employment_tab emp on emp.employee_id = empl.id
join tasks.tasks_tab task on task.employment_id = emp.id
join tasks.projects_tab proj on proj.id = task.project_id
join tasks.steps_tab step on step.task_id = task.id
where task.due_date < coalesce(step.end_date, task.due_date +3);
```

tasks\_track/postgres@PostgreSQL 14

Query Query History Data Output Messages

	employee text	project_name character varying (128)	task_name character varying (128)	exp_days integer
1	Ezra Castillo	project1595	task 25	3
2	Makayla Howard	project 812	task 25	3
3	Vincent Harrison	project1438	task 22	3
4	Aaliyah Boyd	project 1795	task 25	3
5	Emilia Shaw	project166	task 13	3
6	Mila Wells	project 201	task 23	3
7	Josiah Anderson	project810	task 22	3
8	Parker Chapman	project 291	task 23	3
9	Asher Fernandez	project1529	task 22	3
10	Savannah Banks	project 630	task 22	3
11	Zara Gordon	project1739	task 25	3
12	Makayla Reed	project446	task 21	3
13	Chloe Ford	project37	task 14	3
14	Lucas Hayes	project 1616	task 23	3
15	Antonio Powell	project164	task 16	3
16	Ellie Garcia	project 783	task 22	3
17	Savannah Baker	project727	task 23	3
18	Luke Bishop	project168	task 16	3
19	Everly Harris	project644	task 22	3
20	Emily Porter	project567	task 25	3
21	William Martin	project841	task 25	3
22	Genesis Hughes	project531	task 22	3

Total rows: 1000 of 5054 Query complete 00:00:00.499

6 - Результат выполнения 6 запроса

— Составить список проектов, в выполнении которого участвует более трех человек.

```
select proj.project_name, count(empl.id)
  from tasks.projects_tab proj
 join tasks.tasks_tab task on task.project_id = proj.id
 join tasks.steps_tab step on step.task_id = task.id
 join tasks.employment_tab emp on emp.employee_id in (step.employment_id, task.employment_id,
proj.supervisor_id, proj.manager_id)
 join tasks.employees_tab empl on emp.employee_id = empl.id
 group by proj.project_name
 having count(empl.id) > 3;
```

Query Query History Data Output Messages		
	project_name character varying (128)	count bigint
1	project 100	14
2	project 1000	16
3	project 1002	12
4	project 1004	33
5	project 1018	23
6	project 102	15
7	project 1029	20
8	project 1035	8
9	project 1039	9
10	project 1043	9
11	project 1046	17
12	project 1047	22
13	project 1056	37
14	project 1077	16
15	project 1079	11
16	project 1080	12
17	project 1087	7
18	project 1098	4
19	project 1099	18
20	project 11	152
21	project 1105	7
22	project 1113	27
Total rows: 1000 of 1101 Query complete 00:00:29.006		

7 - Результат выполнения 7 запроса

### Задание 3.

— для руководителей проектов, содержащее сведения об исполнителях, отделах, сроках выполнения заданий, включенных в проект.

```
create or replace view tasks.proj_exec_dep_dates as
select empl.name||' '||empl.surname as employee,
       dep.name                    as department_name,
       proj.project_name           as project_name,
       proj.contract_finish_date   as project_due_date,
       task.name                   as task_name,
       task.due_date              as task_due_date
from tasks.projects_tab proj
join tasks.tasks_tab task on task.project_id = proj.id
join tasks.employment_tab emp on emp.employee_id in (task.employment_id)
join tasks.employees_tab empl on emp.employee_id = empl.id
join tasks.departments_tab dep on dep.id = emp.department_id;
```



	employee text	department_name character varying (256)	project_name character varying (128)	project_due_date date	task_name character varying (128)	task_due_date date
1	Addison Peterson	Legal	project1205	2023-12-04	task 24	2023-11-01
2	Naomi Riley	Legal	project 190	2011-12-28	task 13	2010-11-26
3	Kennedy Boyd	Legal	project1650	2009-09-08	task 25	2009-07-01
4	Kennedy Boyd	Legal	project137	2013-12-19	task 14	2013-07-17
5	Violet Nguyen	Legal	project 206	2006-11-21	task 22	2006-07-19
6	Eva Simmons	Legal	project444	2008-12-11	task 22	2008-12-04
7	Eva Simmons	Legal	project 442	2010-12-25	task 23	2010-12-23
8	Eva Simmons	Legal	project216	2014-12-04	task 22	2013-09-01
9	Mason Gomez	Legal	project 38	2020-12-16	task 18	2019-10-07
10	Madison Wheeler	Legal	project1192	2003-12-03	task 23	2002-10-01
11	Madison Wheeler	Legal	project1104	2011-09-07	task 24	2010-01-07
12	Madison Wheeler	Legal	project 113	2020-12-01	task 12	2019-11-01
13	Leah Watkins	Legal	project1426	2025-08-25	task 24	2024-01-16
14	Leah Watkins	Legal	project 1408	2003-05-17	task 22	2003-03-10
15	Leah Watkins	Legal	project1096	2012-12-16	task 22	2012-12-09
16	Claire Lawrence	Legal	project1169	2005-12-23	task 22	2004-12-13
17	Claire Lawrence	Legal	project1158	2004-12-14	task 25	2004-10-07
18	Claire Lawrence	Legal	project1131	2009-12-19	task 25	2008-09-18
19	Claire Lawrence	Legal	project372	2009-12-01	task 21	2008-09-01
20	Levi Harvey	Legal	project1495	2010-12-18	task 24	2010-12-10
21	Levi Harvey	Legal	project993	2011-09-06	task 21	2010-01-01
22	Levi Harvey	Legal	project448	2016-12-13	task 21	2015-11-10
Total rows: 1000 of 2324		Query complete 00:00:00.381				

## 8 - Данные из первого представления

— список проектов, срок выполнения которых истекает сегодня и которые включают больше трех невыполненных заданий.

```
create or replace view tasks.due_projects as
select proj.project_name, count(1)
from tasks.projects_tab proj
join tasks.tasks_tab task on task.project_id = proj.id
where task.check_date is null or task.status != 'Done'
group by proj.project_name
having count(1) > 3;
```

Query Query History Data Output Messages			
	project_name character varying (128)	count bigint	
1	project402	5	
2	project 151	4	
3	project 1258	5	
4	project 1616	4	
5	project208	4	
6	project 417	4	
7	project60	6	
8	project409	4	
9	project24	8	
10	project 1400	4	
11	project268	4	
12	project416	4	
13	project1438	4	
14	project517	4	
15	project 1798	5	
16	project 1674	4	
17	project74	5	
18	project1233	5	
19	project858	5	
20	project1522	5	
21	project 442	4	
22	project 25	6	
Total rows: 721 of 721		Query complete 00:00:00.535	

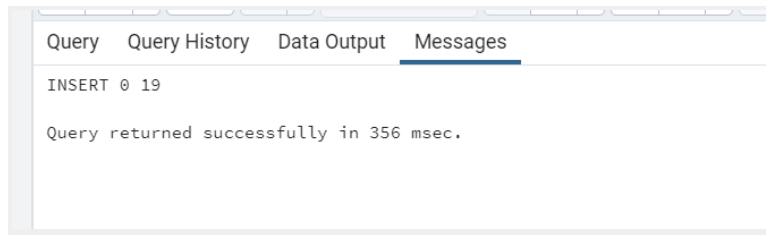
9 - Данные из второго представления

**Запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.**

**— Insert для новых проектов, для которых нужна декомпозиция задач**

```
insert into tasks.steps_tab(task_id, employment_id, status, name, step_cost)
select pre.task_id, pre.manager_id, 'Not_started', 'Initial management', 5000
from (select proj.id as proj_id, proj.manager_id, task.id as task_id
      from tasks.projects_tab proj
      join tasks.tasks_tab task on task.project_id = proj.id
      where proj.id in (select proj.id
                       from tasks.projects_tab proj
                       join tasks.tasks_tab task on task.project_id = proj.id
                       group by proj.id
                       having count(distinct task.id) = 1
                       )) pre
```

Время выполнения: 356мс

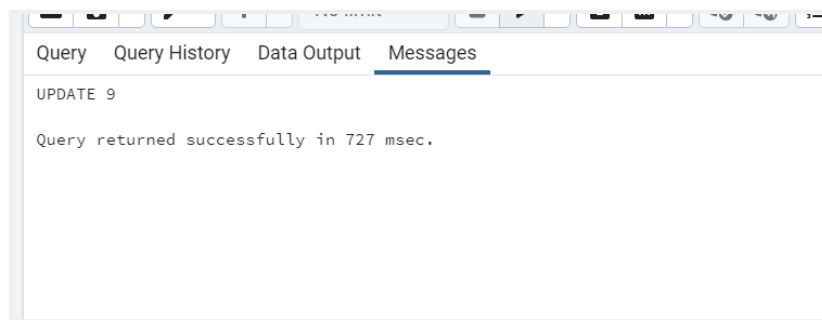


10 - скриншот 1

### -- Update статуса для проектов, в которых все задания сделаны

```
update tasks.projects_tab
set status = 'Done'
where id in (
    select proj.id as proj_id
    from tasks.projects_tab proj
    join tasks.tasks_tab task on task.project_id = proj.id
    where task.status = 'Done'
    group by proj.id
    having count(1) = (
        select count(1)
        from tasks.projects_tab projc
        join tasks.tasks_tab taskc on taskc.project_id = proj.id
        where projc.id = proj.id
    )
);
```

Время выполнения: 727мс



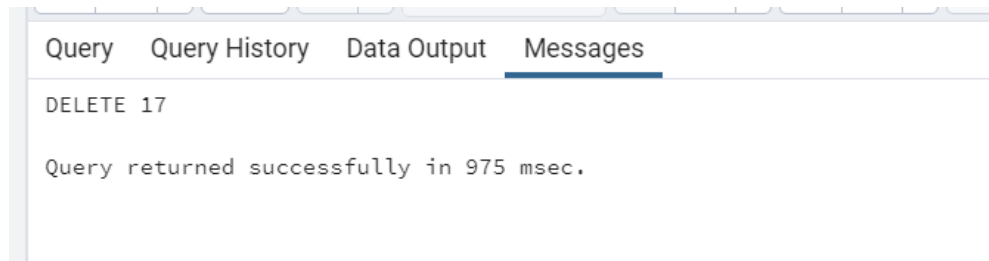
11 - Скриншот 2

### — DELETE для старых замороженных проектов

```
delete from
tasks.projects_tab
where id in (
    select proj.id as proj_id
    from tasks.projects_tab proj
    join tasks.tasks_tab task on task.project_id = proj.id
    where task.status = 'Frozen'
    group by proj.id
    having count(1) = (
        select count(1)
        from tasks.projects_tab projc
        join tasks.tasks_tab taskc on taskc.project_id = projc.id
        where projc.id = proj.id
    )
);
```

```
and contract_finish_date < current_date - 180
);
```

Время выполнения: 975 мс



12 - Скриншот 3

### Создадим индексы и замерим выполнение с ними:

В силу наличия ограничений, индексы на РК полях созданы постгресом автоматически, для их дропа придется дропнуть и ограничение, что проблематично, а также большинство запросов построены, в основном, на айдишниках-РК или полях с низкой селективностью в запросах (Например, по статусу), что делает использование индексов неэффективным

The screenshot shows the 'Explain' window in a database client. The 'Analysis' tab is selected, displaying a table with 5 columns: Step, Operation, Rows, and Loops. The table lists 13 steps of the query execution plan. Steps 5, 6, 7, 8, 9, 10, 11, 12, and 13 are visible. The operations include Hash Inner Join, Seq Scan, Hash, Seq Scan, Aggregate, Nested Loop Inner Join, Index Only Scan, and Index Scan. The rows and loops for each step are also listed.

Step	Operation	Rows	Loops
5.	→ Hash Inner Join (rows=1391 loops=1) Hash Cond: (task.project_id = proj.id)	1391	1
6.	→ Seq Scan on tasks_tab as task (rows=1692 loops=1) Filter: ((status)::text = 'Frozen'::text) Rows Removed by Filter: 4904	1692	1
7.	→ Hash (rows=1549 loops=1) Buckets: 2048 Batches: 1 Memory Usage: 71 kB	1549	1
8.	→ Seq Scan on projects_tab as proj (rows=1549 loops=1) Filter: (contract_finish_date < (CURRENT_DATE - 180)) Rows Removed by Filter: 280	1549	1
9.	→ Aggregate (rows=1 loops=835)	1	835
10.	→ Nested Loop Inner Join (rows=5 loops=835)	5	835
11.	→ Index Only Scan using projects_tab_pkey on projects_tab as projc (r...) Index Cond: (id = proj.id)	1	835
12.	→ Seq Scan on tasks_tab as taskc (rows=5 loops=835) Filter: (project_id = proj.id) Rows Removed by Filter: 6591	5	835
13.	→ Index Scan using projects_tab_pkey on projects_tab as projects_tab (rows=0 loops=0) Index Cond: (id = 'ANY_subquery'.proj_id)	0	0

13 - Использование оптимайзером РК-индексов

Поэтому рассмотрим использование индексов на самом долгом запросе из всех:

Запрос 7 выполняется целую минуту!

Query Query History Data Output Messages			
	project_name	count	
	character varying (128)	bigint	
1	project 100	14	
2	project 1000	16	
3	project 1002	12	
4	project 1004	33	
5	project 1018	23	
6	project 102	15	
7	project 1029	20	
8	project 1035	8	
9	project 1039	9	
10	project 1043	9	
11	project 1046	17	
12	project 1047	22	
13	project 1056	37	
14	project 1077	16	
15	project 1079	11	
16	project 1080	12	
17	project 1087	7	
18	project 1098	4	
19	project 1099	18	
20	project 11	152	
21	project 1105	7	
Total rows: 1000 of 1086		Query complete 00:01:12.405	

#### 14 - Время выполнения запроса 7 без индекса

Создадим индекс на employment\_tab:

```
create index on tasks.employment_tab(employee_id);
```

	project_name character varying (128)	count bigint
1	project 1784	4
2	project564	11
3	project1816	7
4	project558	16
5	project1495	8
6	project402	10
7	project 151	20
8	project1739	36
9	project1703	15
10	project171	13
11	project1435	9
12	project1642	7
13	project661	5
14	project1106	21
15	project1530	12
16	project1031	14
17	project776	15
18	project1602	16
19	project 129	91
20	project208	19
21	project545	13
Total rows: 1000 of 1086		Query complete 00:00:00.349

15 - Запрос после создания индекса

Теперь запрос выполняется меньше секунды.

Докажем неэффективность индекса на полях, используемых в запросам с низкой селективностью:

```
create or replace view tasks.due_projects as
select proj.project_name, count(1)
  from tasks.projects_tab proj
 join tasks.tasks_tab task on task.project_id = proj.id
 where task.check_date is null or task.status != 'Done'
 group by proj.project_name
 having count(1) > 3;
```

Селект из представление занимает 72мс

Создание индекса:

```
create index on tasks.tasks_tab(status);
```

Теперь запрос выполняется в 2 раза дольше.

tasks\_tab/postgres@postgres14

Query Query History Data Output Messages

	project_name character varying (128)	count bigint
1	project558	4
2	project402	5
3	project 151	4
4	project1703	5
5	project171	5
6	project1642	4
7	project1106	5
8	project 1258	5
9	project 1616	4
10	project 129	4
11	project208	4
12	project 417	4
13	project60	6
14	project1161	4
15	project1322	4
16	project 251	4
17	project37	8
18	project713	4
19	project409	4
20	project24	8
21	project 1400	4

Total rows: 721 of 721 Query complete 00:00:00.139

16 - запрос с индексом

Дропнем индекс.

```
drop index tasks.tasks_tab_status_idx;
```

Query   Query History   Data Output   Messages			
<div> <div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> </div>			
	project_name character varying (128) 🔒	count bigint 🔒	
1	project558	4	
2	project402	5	
3	project 151	4	
4	project1703	5	
5	project171	5	
6	project1642	4	
7	project1106	5	
8	project 1258	5	
9	project 1616	4	
10	project 129	4	
11	project208	4	
12	project 417	4	
13	project60	6	
14	project1161	4	
15	project1322	4	
16	project 251	4	
17	project37	8	
18	project713	4	
19	project409	4	
20	project24	8	
21	project 1400	4	
Total rows: 721 of 721		Query complete 00:00:00.069	

17 - время вернулось в норму

### Вывод:

В процессе выполнения данной лабораторной работы я овладела практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL с использованием подзапросов при модификации данных и индексов.