

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»**

**Выполнила:**  
студентка II курса ИКТ  
группы К3243  
Костень Анна Сергеевна

**Проверила:**  
Говорова Марина Михайловна

Санкт-Петербург  
2021

**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:**

Наименование БД – «БД Контроль выполнения заданий». Схема логической модели БД, сгенерированная в Generate ERD:

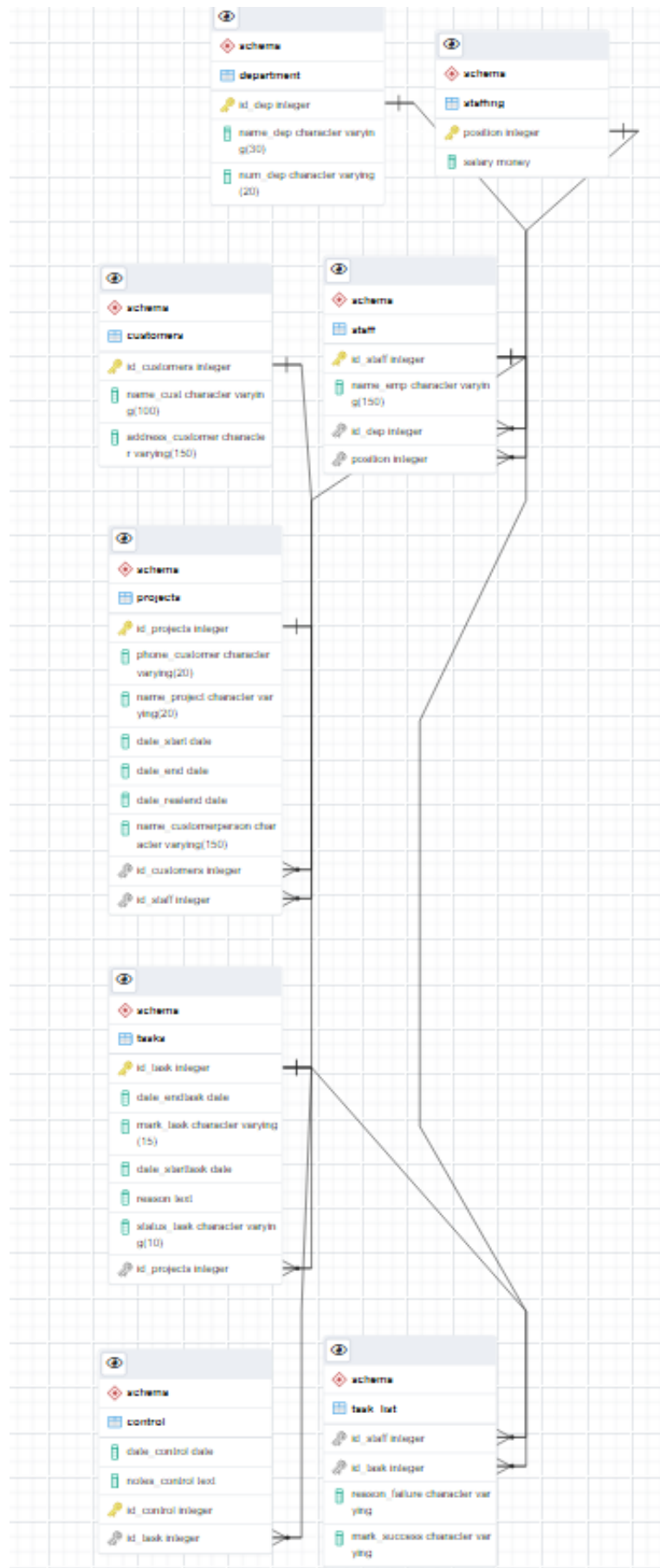


Рисунок 1 – Схема, сгенерированная в Generate ERD

## Задание 1. Создайте запросы:

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

### Листинг 1 – Первый запрос

```
select
  tasks.id_task as "Задание",
  projects.name_project as "Проект",
  customers.name_cust as "Организация",
  department.name_dep as "Отдел",
  staff.name_emp as "Исполнители"
from
  schema.customers, schema.department, schema.tasks, schema.staff,
schema.projects
where
  projects.id_projects = tasks.id_projects and customers.id_customers =
projects.id_customers and department.id_dep = staff.id_dep
order by tasks.id_task
```

	Задание integer	Проект character varying (20)	Организация character varying (100)	Отдел character varying (30)	Исполнители character varying (150)
1	1	успешный проект	ОАО Два по БД	отдел хорошего настроения	Титенко Елена
2	1	успешный проект	ОАО Два по БД	отдел терпил	Махнева Анастасия
3	1	успешный проект	ОАО Два по БД	глиномесный цех	Смирнов Тимур Олегович
4	1	успешный проект	ОАО Два по БД	столовая	Асонов Николай Павлович
5	1	успешный проект	ОАО Два по БД	офис психологической помощи	Фатин Леша)
6	2	успешный проект	ООО Помогите	отдел хорошего настроения	Титенко Елена
7	2	успешный проект	ООО Помогите	отдел терпил	Махнева Анастасия
8	2	успешный проект	ООО Помогите	глиномесный цех	Смирнов Тимур Олегович
9	2	успешный проект	ООО Помогите	столовая	Асонов Николай Павлович
10	2	успешный проект	ООО Помогите	офис психологической помощи	Фатин Леша)
11	3	успешный проект	ОАО Два по БД	отдел хорошего настроения	Титенко Елена
12	3	успешный проект	ОАО Два по БД	отдел терпил	Махнева Анастасия
13	3	успешный проект	ОАО Два по БД	глиномесный цех	Смирнов Тимур Олегович
14	3	успешный проект	ОАО Два по БД	столовая	Асонов Николай Павлович
15	3	успешный проект	ОАО Два по БД	офис психологической помощи	Фатин Леша)
16	4	успешный проект	ООО Помогите	отдел хорошего настроения	Титенко Елена
17	4	успешный проект	ООО Помогите	отдел терпил	Махнева Анастасия
18	4	успешный проект	ООО Помогите	глиномесный цех	Смирнов Тимур Олегович
19	4	успешный проект	ООО Помогите	столовая	Асонов Николай Павлович
20	4	успешный проект	ООО Помогите	офис психологической помощи	Фатин Леша)

Рисунок 1 – результат выполнения первого запроса (первые 20 строк)

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

## Листинг 2 – Второй запрос

```
select
  projects.name_project
from
  schema.projects
where
  projects.date_start < current_date - interval '1 month'
```

	name_project character varying (20) 🔒
1	успешный проект
2	успешный проект
3	успешный проект
4	успешный проект
5	успешный проект
6	успешный проект
7	проект1
8	проект2
9	проект3
10	проект4
11	проект5
12	haha factory

Рисунок 2 – результат выполнения второго запроса

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

## Листинг 3 – Третий запрос

```
select
  staff.name_emp,
  staff.id_dep
from
  schema.staff, schema.staffing
where
  staffing.position = staff.position
  and (select avg(salary) from schema.staffing) < staffing.salary
order by staff.id_dep
```

	name_emp character varying (150)	id_dep integer
1	Титенко Елена	1
2	Смирнов Тимур Олегович	2
3	Асонов Николай Павлович	3

Рисунок 3 – Результат выполнения третьего запроса

Найти отдел, работающий над максимальным количеством проектов.

#### Листинг 4 – Четвертый запрос

```
select "dep"
from
(
  select
    staff.id_dep,
    count(distinct projects.id_staff),
    department.name_dep as dep
  from
    schema.staff,
    schema.department,
    schema.projects
  where
    staff.id_dep = department.id_dep and
    projects.id_staff = staff.id_staff
  group by staff.id_dep, department.name_dep
  order by count(distinct projects.id_staff) desc
) as tbl1
where
  count = (
    select max(count)
    from
      (
        select
          staff.id_dep,
          count(distinct projects.id_staff),
          department.name_dep as dep
        from
          schema.staff,
          schema.department,
          schema.projects
        where
          staff.id_dep = department.id_dep and
```

```

    projects.id_staff = staff.id_staff
  group by staff.id_dep, department.name_dep
  order by count(distinct projects.id_staff) desc
) as tbl2
)

```


	dep	
	character varying (30)	
1	столовая	
2	офис психологической помощи	

Рисунок 4 – Результат выполнения четвертого запроса

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

Листинг 5 – пятый запрос

```

select
  staff.name_emp,
  projects.name_project,
  tasks.id_task,
  tasks.date_endtask
from
  schema.staff
  inner join schema.projects on staff.id_staff = projects.id_staff
  left join schema.tasks on projects.id_projects = tasks.id_projects

```

Для выполнения данного задания данных, добавленных вручную оказалось недостаточно, поэтому на этом этапе были добавлены дополнительные данные в таблицу projects.

	<b>name_emp</b> character varying (150)	<b>name_project</b> character varying (20)	<b>id_task</b> integer	<b>date_endtask</b> date
1	Фатин Леша)	успешный проект	1	2020-12-12
2	Фатин Леша)	успешный проект	2	2020-12-12
3	Фатин Леша)	успешный проект	3	2020-12-12
4	Фатин Леша)	успешный проект	4	2020-12-12
5	Фатин Леша)	успешный проект	5	2020-12-12
6	Титенко Елена	проект2	[null]	[null]
7	Смирнов Тимур Олегович	проект5	[null]	[null]
8	Махнева Анастасия	проект3	[null]	[null]
9	Фатин Леша)	успешный проект	[null]	[null]
10	Асонов Николай Павлович	успешный проект	[null]	[null]
11	Фатин Леша)	успешный проект	[null]	[null]
12	Фатин Леша)	успешный проект	[null]	[null]
13	Махнева Анастасия	проект4	[null]	[null]
14	Махнева Анастасия	проект1	[null]	[null]

Рисунок 5 – Результат выполнения пятого запроса

Можно заметить, что некоторым сотрудникам на данный момент не передано ни одного задания.

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

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

Листинг 6 – Седьмой запрос

```
select
  projects.id_projects,
  projects.name_project
from
  schema.projects,
  schema.staff
where
  projects.id_staff = staff.id_staff
```



```
group by projects.id_projects, projects.name_project
having count(staff.id_staff)>3
```

Таких проектов в моей БД нет.

	id_projects [PK] integer	name_project character varying (20)

Рисунок 6 – Результат выполнения седьмого запроса

## Задание 2. Создать представления:

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

### Листинг 8 – Создание первого представления

```
create view schema.team_leads as
select
    department.name_dep as "отдел",
    staff.name_emp as "исполнитель",
    projects.name_project as "название проекта",
    tasks.id_task as "задание",
    tasks.date_endtask as "дедлайн"
from
    schema.department,
    schema.staff,
    schema.projects,
    schema.tasks
where
    department.id_dep = staff.id_dep and
    projects.id_staff = staff.id_staff and
    tasks.id_projects = projects.id_projects
order by tasks.id_task
```

	отдел character varying (30)	исполнитель character varying (150)	название проекта character varying (20)	задание integer	дедлайн date
1	офис психологической помощи	Фатин Леша)	успешный проект	1	2020-12-12
2	офис психологической помощи	Фатин Леша)	успешный проект	2	2020-12-12
3	офис психологической помощи	Фатин Леша)	успешный проект	3	2020-12-12
4	офис психологической помощи	Фатин Леша)	успешный проект	4	2020-12-12
5	офис психологической помощи	Фатин Леша)	успешный проект	5	2020-12-12
6	столовая	Асонов Николай Павлович	успешный проект	7	2022-04-01

Рисунок 8 – Представление team\_leads

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

Для данного представления в таблицу были добавлены дополнительные данные.

#### Листинг 9 – Создание второго представления

```
create view schema.troubles_of_the_day as
select
  projects.id_projects,
  projects.name_project
from
  schema.projects,
  schema.tasks
where
  projects.id_projects = tasks.id_projects and
  projects.date_end = current_date and
  tasks.status_task = 'в работе'
group by projects.id_projects, projects.name_project
having count(tasks.id_task)>3
```

1	select * from schema.troubles_of_the_day			
Результат		План выполнения	Сообщения	Notifications
	<b>id_projects</b> integer		<b>name_project</b> character varying (20)	
1	16		haha factory	

Рисунок 9 – Представление troubles\_of\_the\_day

#### Задание 3. Модификация данных.

Запрос с insert.

Добавить задание, начинающееся сегодня, в проект, относящийся к определенному проекту.

	<b>id_task</b> [PK] integer	<b>date_endtask</b> date	<b>mark_task</b> character varying (15)	<b>date_starttask</b> date	<b>reason</b> text	<b>status_task</b> character varying (10)	<b>id_projects</b> integer
1	1	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
2	2	2020-12-12	отметка	2019-12-12	не получилось ничего:(	В процессе	6
3	3	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
4	4	2020-12-12	отметка	2019-12-12	не получилось ничего:(	В процессе	6
5	5	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
6	7	2022-04-01		2022-01-02		выполнено	4
7	8	2022-04-06		2021-01-09		в работе	16
8	9	2022-04-06		2021-02-07		в работе	16
9	10	2022-04-06		2021-03-01		в работе	16
10	11	2022-05-01		2022-01-07		в работе	16

Рисунок 10 – Таблица заданий до выполнения запроса

## Листинг 10 – Запрос с insert

```
insert into schema.tasks (date_endtask, mark_task, date_starttask, reason,
status_task, id_projects)
values ('2022-06-01', '', current_date, '', 'в работе',
(select projects.id_projects from schema.projects where
projects.name_project = 'haha factory'))
```

Результат	План выполнения	Сообщения	Notifications		<b>id_task</b> [PK] integer	<b>date_endtask</b> date	<b>mark_task</b> character varying (15)	<b>date_starttask</b> date	<b>reason</b> text	<b>status_task</b> character varying (10)	<b>id_projects</b> integer
				1	1	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
				2	2	2020-12-12	отметка	2019-12-12	не получилось ничего:(	В процессе	6
				3	3	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
				4	4	2020-12-12	отметка	2019-12-12	не получилось ничего:(	В процессе	6
				5	5	2020-12-12		2019-12-12	не получилось ничего:(	Завершен	2
				6	7	2022-04-01		2022-01-02		выполнено	4
				7	8	2022-04-06		2021-01-09		в работе	16
				8	9	2022-04-06		2021-02-07		в работе	16
				9	10	2022-04-06		2021-03-01		в работе	16
				10	11	2022-05-01		2022-01-07		в работе	16
				11	12	2022-06-01		2022-04-06		в работе	16

Рисунок 11 – Таблица заданий после выполнения запроса

Запрос с update.

Добавить повысить до позиции с максимальной зарплатой сотрудника, занятого больше, чем в 4 проектах.






Результат	План выполнения		Сообщения	Notifications	
 id_staff [PK] integer 		name_emp character varying (150) 		id_dep integer 	position integer 
1	2	Титенко Елена		1	13
2	3	Махнева Анастасия		5	11
3	4	Смирнов Тимур Олегович		2	2
4	5	Асонов Николай Павлович		3	2
5	6	Фатин Леша)		4	11

Рисунок 12 – Таблица сотрудников до выполнения запроса

#### Листинг 11 – Запрос с update

```
update schema.staff
set position = (select staffing.position from schema.staffing where salary =
(select max(salary) from schema.staffing))
where
  staff.id_staff = (select projects.id_staff
                    from schema.projects
                    group by projects.id_staff
                    having count(projects.id_projects) > 4)
```

Результат	План выполнения		Сообщения	Notifications	
 id_staff [PK] integer 		name_emp character varying (150) 		id_dep integer 	position integer 
1	2	Титенко Елена		1	13
2	3	Махнева Анастасия		5	11
3	4	Смирнов Тимур Олегович		2	2
4	5	Асонов Николай Павлович		3	2
5	6	Фатин Леша)		4	13

Рисунок 13 – Таблица сотрудников после выполнения запроса

Запрос с delete.

Удалить заказчиков, которым не принадлежит ни один проект.

	<b>id_customers</b> [PK] integer	<b>name_cust</b> character varying (100)	<b>address_customer</b> character varying (150)
1	1	ООО Шаражкина контора	ул. Пушкина, д. Колотушкина
2	2	ООО Спёрбанк	ул. Улица, д. 666
3	3	ООО Помогите	ул. Допсы, д. Отчисления
4	4	ОАО Два по БД	Биржевой, 16
5	5	ООО Сакичарм	ул. Хорошая, д. 17

Рисунок 14 – Таблица заказчиков до выполнения запроса

#### Листинг 12 – Запрос с delete

```
delete from schema.customers  
where id_customers not in (select projects.id_customers from schema.projects)
```

	<b>id_customers</b> [PK] integer	<b>name_cust</b> character varying (100)	<b>address_customer</b> character varying (150)
1	1	ООО Шаражкина контора	ул. Пушкина, д. Колотушкина
2	3	ООО Помогите	ул. Допсы, д. Отчисления
3	4	ОАО Два по БД	Биржевой, 16

Рисунок 15 – Таблица заказчиков после выполнения запроса

#### Задание 4. История запросов

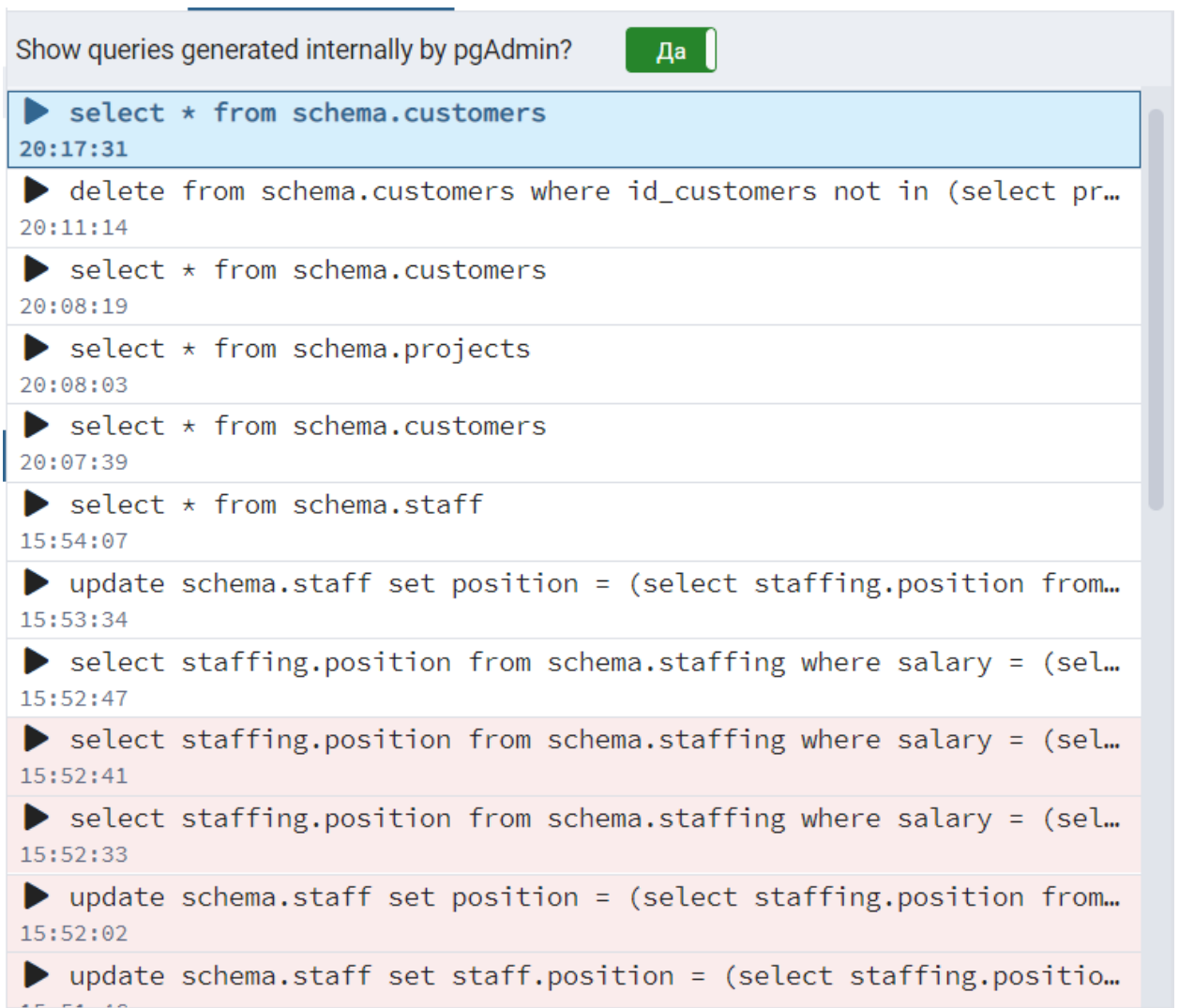


Рисунок 16 – История запросов из Query Tool

### Задание 5. Explain.

Рассмотрим выполнение запроса (листинг 2) с использованием explain.

	QUERY PLAN	
	text	
1	Seq Scan on projects (cost=0.00..12.80 rows=53 width=58)	
2	[...] Filter: (date_start < (CURRENT_DATE - '1 mon'::interval))	

Рисунок 17 – Запрос с листинга 3 с explain

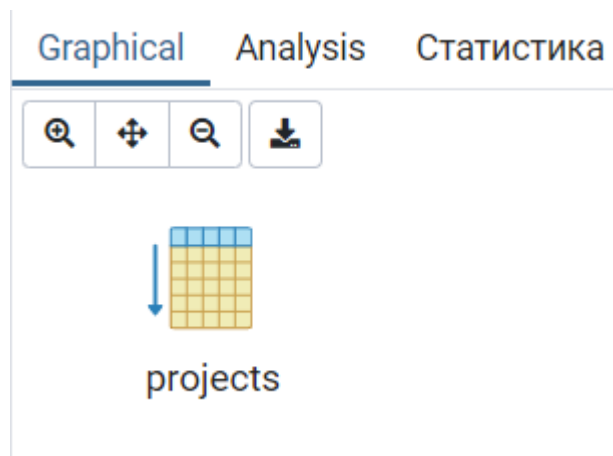


Рисунок 18 – Графическое представление запроса

Время выполнения запроса с explain – 92 мс, время генерации графического представления – 82 мс.

Далее я выполнила запрос с листинга 3.

	<b>QUERY PLAN</b> text	
1	Sort (cost=17.24..17.42 rows=73 width=322)	
2	[...] Sort Key: staff.id_dep	
3	[...] InitPlan 1 (returns \$0)	
4	[...] -> Aggregate (cost=1.08..1.09 rows=1 width=32)	
5	[...] -> Seq Scan on staffing staffing_1 (cost=0.00..1.06 rows=6 width=32)	
6	[...] -> Hash Join (cost=1.10..13.89 rows=73 width=322)	
7	[...] Hash Cond: (staff."position" = staffing."position")	
8	[...] -> Seq Scan on staff (cost=0.00..12.20 rows=220 width=326)	
9	[...] -> Hash (cost=1.08..1.08 rows=2 width=4)	
10	[...] -> Seq Scan on staffing (cost=0.00..1.08 rows=2 width=4)	
11	[...] Filter: (\$0 < salary)	

Рисунок 19 – Результат выполнения запроса с листинга 3 с explain

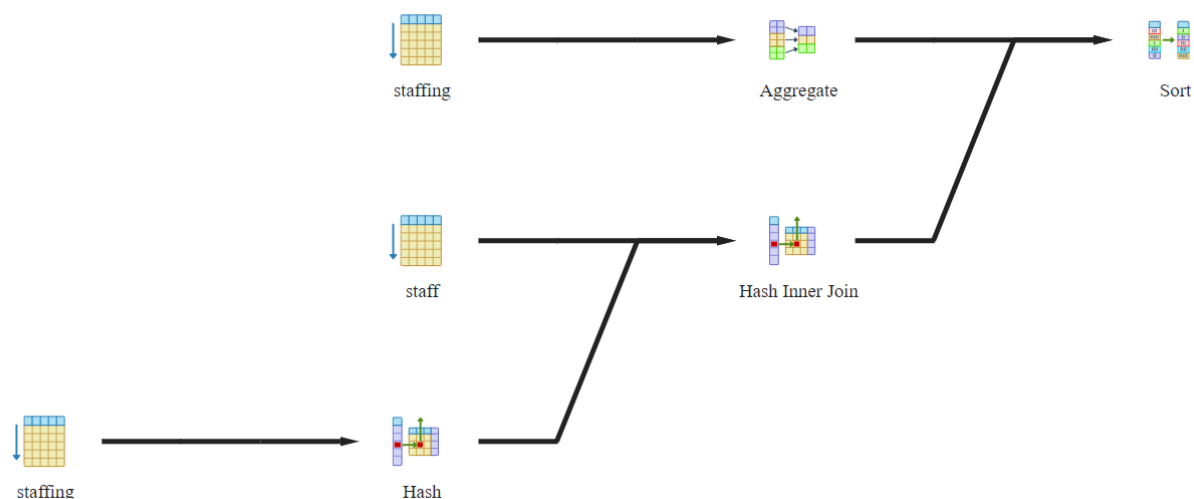


Рисунок 20 – Графическое представление запроса с листинга 3

Время выполнения запроса с explain – 92 мс, время генерации графического представления – 188 мс.

Создание индексов.

Листинг 13 – Создание простого индекса для первого запроса

```
create index idx on schema.projects(id_projects)
```

QUERY PLAN		
	text	
1	Seq Scan on projects (cost=0.00..1.21 rows=4 width=58)	
2	[...] Filter: (date_start < (CURRENT_DATE - '1 mon'::interval))	

Рисунок 21 – План выполнения первого запроса после создания индекса

Время выполнения с индексом – 80 мс.

Листинг 14 – Создание составного индекса для второго запроса

```
create index idx1 on schema.staff(id_staff, name_emp)
```



	QUERY PLAN	
	text	
1	Sort (cost=3.27..3.28 rows=2 width=322)	
2	[...] Sort Key: staff.id_dep	
3	[...] InitPlan 1 (returns \$0)	
4	[...] -> Aggregate (cost=1.08..1.09 rows=1 width=32)	
5	[...] -> Seq Scan on staffing staffing_1 (cost=0.00..1.06 rows=6 width=32)	
6	[...] -> Hash Join (cost=1.10..2.17 rows=2 width=322)	
7	[...] Hash Cond: (staff."position" = staffing."position")	
8	[...] -> Seq Scan on staff (cost=0.00..1.05 rows=5 width=326)	
9	[...] -> Hash (cost=1.08..1.08 rows=2 width=4)	
10	[...] -> Seq Scan on staffing (cost=0.00..1.08 rows=2 width=4)	
11	[...] Filter: (\$0 < salary)	

Рисунок 22 – План выполнения второго запроса после создания индекса

Время выполнения с индексом – 83 мс.

Удаление индексов.

Для начала убедимся в их наличии.

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	schema	projects	projects_pkey	[null]	CREATE UNIQUE INDEX projects_pkey ON schema.projects USING btree (id_projects)
2	schema	projects	r11	[null]	CREATE UNIQUE INDEX r11 ON schema.projects USING btree (id_projects)
3	schema	projects	idx	[null]	CREATE INDEX idx ON schema.projects USING btree (id_projects)

Рисунок 23 – Индексы таблицы projects

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	schema	staff	staff_pkey	[null]	CREATE UNIQUE INDEX staff_pkey ON schema.staff USING btree (id_staff)
2	schema	staff	r13	[null]	CREATE UNIQUE INDEX r13 ON schema.staff USING btree (id_staff)
3	schema	staff	idx1	[null]	CREATE INDEX idx1 ON schema.staff USING btree (id_staff, name_emp)

Рисунок 24 – Индексы таблицы staff

Листинг 15 – Пример удаления индекса

```
drop index schema.idx1
```



	 schemaname name	 tablename name	 indexname name	 tablespace name	 indexdef text
1	schema	staff	staff_pkey	[null]	CREATE UNIQUE INDEX staff_pkey ON schema.staff USING btree (id_staff)
2	schema	staff	r13	[null]	CREATE UNIQUE INDEX r13 ON schema.staff USING btree (id_staff)

Рисунок 25 – Индексы таблицы staff после удаления idx1

**Вывод:** SQL – запросы позволяют не только находить необходимые данные по определенным условиям, но и изменять их, дополнять и удалять. Я узнала об индексах, их создании и применении, однако в моей базе данных они не показали достаточной эффективности в связи с небольшим количеством данных.