

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

О Т Ч Е Т

по дисциплине «Проектирование и реализация баз данных»

на тему: Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL

Обучающийся: Олейникова Полина Леонидовна, К32402

Преподаватель: Говорова М. М.

Дата 17.04.2023

ВВЕДЕНИЕ

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.

Индивидуальное задание: Вариант 4. БД «Учет выполнения заданий»

Описание предметной области: Компания осуществляет деятельность по выполнению проектов на заказ. По каждому проекту составляется договор с Заказчиком (в 2-х экземплярах для каждой стороны). По каждому договору оформляется два счета - на предоплату и остаток. После выполнения проекта подписывается Акт выполненных работ (в 2-х экземплярах для каждой стороны). Каждый договор сопровождается менеджером со стороны компании, который обеспечивает коммуникации между заказчиком и руководителем проекта.

Проекты состоят из нескольких заданий (этапов), каждый из которых имеет свою стоимость в рамках всего проекта. Для каждого задания проекта руководитель проекта составляет график контроля выполнения, включающий несколько дат для каждого задания. По итогу контроля хранится информация

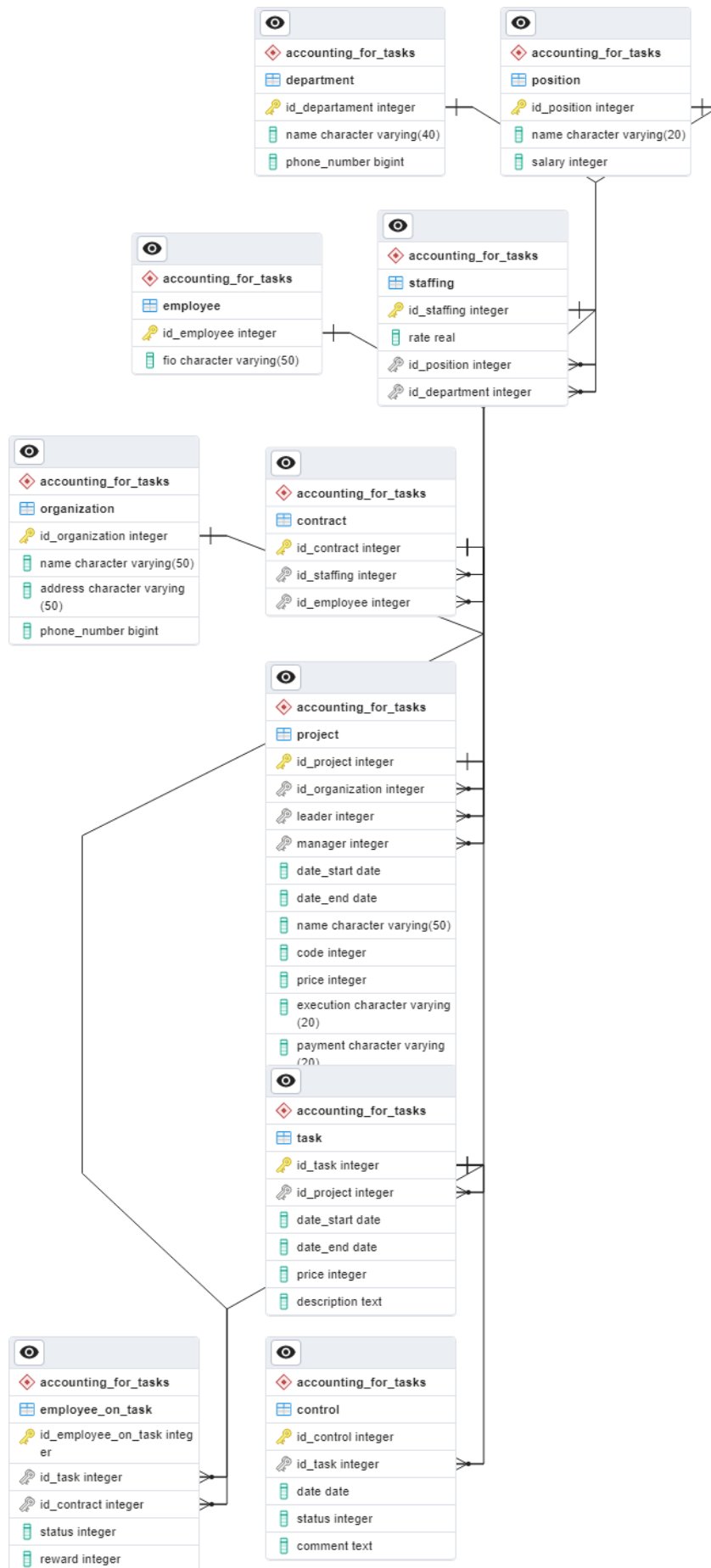
о выполнении к дате контроля задания (в процентах), поясняющий комментарий о причинах невыполнения или отставания выполнения задания.

Каждый проект имеет руководителя проекта из числа сотрудников. Каждый сотрудник может участвовать в одном или нескольких проектах, или временно не участвовать ни в каких проектах. Над каждым проектом может работать несколько сотрудников отделов, или временно проект может быть приостановлен, тогда над ним не работает ни один сотрудник. Над каждым заданием (этапом) в проекте может работать несколько сотрудников. Для участия в проекте с каждым сотрудником заключается договор на выполнение проектных работ.

Каждый сотрудник числится в одном отделе по основной должности согласно штатному расписанию отдела. Сотрудник может работать в другом отделе на условиях штатного совмещения на 0,5 ставки.

БД должна содержать следующий минимальный набор сведений: Номер сотрудника. Фамилия сотрудника. Имя сотрудника. Отчество сотрудника. Должность сотрудника. Оклад сотрудника. Название организации-заказчика. Номер организации. Адрес организации. Контактное лицо и его контакты. Номер телефона отдела. Номер отдела. Название отдела. Код проекта. Название проекта. Сроки выполнения проекта. Руководитель проекта. Статус выполнения. Статус оплаты. Номер задания. Дата начала выполнения задания. Срок выполнения задания. Статус выполнения задания. Вознаграждение за задание сотруднику. Статус выполнения задания каждым сотрудником. Дата контроля выполнения задания. Причина невыполнения или отставания выполнения задания.

Схема логической модели базы данных:



1 Выполнение

1.1 Запросы

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

```
SELECT task.id_task, task.description as "task", project.name as "project", organization.name as
"organization", department.name as "department", employee.fio as "employee"
FROM accounting_for_tasks.task
JOIN accounting_for_tasks.project using(id_project)
JOIN accounting_for_tasks.organization ON project.id_organization = organization.id_organization
JOIN accounting_for_tasks.employee_on_task using(id_task)
JOIN accounting_for_tasks.contract ON employee_on_task.id_contract = contract.id_contract
JOIN accounting_for_tasks.employee ON contract.id_employee = employee.id_employee
JOIN accounting_for_tasks.staffing ON contract.id_staffing = staffing.id_staffing
JOIN accounting_for_tasks.department ON staffing.id_department = department.id_department
```

Query

Query History

Scratch Pad

```
1 SELECT task.id_task, task.description as "task", project.name as "project",
2 organization.name as "organization", department.name as "department",
3 employee.fio as "employee"
4 FROM accounting_for_tasks.task
5 JOIN accounting_for_tasks.project using(id_project)
6 JOIN accounting_for_tasks.organization ON project.id_organization =
7 organization.id_organization
8 JOIN accounting_for_tasks.employee_on_task using(id_task)
9 JOIN accounting_for_tasks.contract ON employee_on_task.id_contract =
10 contract.id_contract
11 JOIN accounting_for_tasks.employee ON contract.id_employee =
12 employee.id_employee
13 JOIN accounting_for_tasks.staffing ON contract.id_staffing =
14 staffing.id_staffing
15 JOIN accounting_for_tasks.department ON staffing.id_department =
16 department.id_department
```

Data Output

Messages

Notifications

	id_task integer	task text	project character varying (50)	organization character varying (50)	department character varying (40)	employee character varying (50)
1	1	страница персоналий	my.itmo	ИТМО	ДИС	Иванов Иван Иванович
2	2	страница расписания	my.itmo	ИТМО	ДИС	Олейникова Полина Леонидовна

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

```
SELECT project.id_project, project.name FROM accounting_for_tasks.project
WHERE date_start < (NOW()-INTERVAL '1 month')
```

Query		Query History
1	<code>SELECT project.id_project, project.name FROM accounting_for_tasks.project</code>	
2	<code>WHERE date_start < (NOW()-INTERVAL '1 month')</code>	

Data Output		Messages	Notifications
	<code>id_project</code> [PK] integer	<code>name</code> character varying (50)	
1	1	my.itmo	

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

```
SELECT contract.id_contract, employee.fio, department.name as "department", position.salary as
"Зарплата"
```

```
FROM accounting_for_tasks.contract
```

```
JOIN accounting_for_tasks.employee ON employee.id_employee = contract.id_employee
```

```
JOIN accounting_for_tasks.staffing ON staffing.id_staffing = contract.id_staffing
```

```
JOIN accounting_for_tasks.position ON staffing.id_position = position.id_position
```

```
JOIN accounting_for_tasks.department ON department.id_department = staffing.id_department
```

```
WHERE position.salary > (
```

```
SELECT AVG(position.salary)
```

```
FROM accounting_for_tasks.staffing
```

```
JOIN accounting_for_tasks.position ON staffing.id_position = position.id_position
```

```
where staffing.id_department = department.id_department)
```

Query

Query History

Sc

```

1 SELECT contract.id_contract, employee.fio, department.name as "department", position.salary as
2 FROM accounting_for_tasks.contract
3 JOIN accounting_for_tasks.employee ON employee.id_employee = contract.id_employee
4 JOIN accounting_for_tasks.staffing ON staffing.id_staffing = contract.id_staffing
5 JOIN accounting_for_tasks.position ON staffing.id_position = position.id_position
6 JOIN accounting_for_tasks.department ON department.id_department = staffing.id_department
7 WHERE position.salary > (
8 SELECT AVG(position.salary)
9 FROM accounting_for_tasks.staffing
10 JOIN accounting_for_tasks.position ON staffing.id_position = position.id_position
11 where staffing.id_department = department.id_department
12 )
13

```

Data Output

Messages

Notifications

	id_contract integer	fio character varying (50)	department character varying (40)	Зарплата integer
1	2	Олейникова Полина Леонидовна	ДИС	150000
2	3	Сергеев Сергей Сергеевич	ДИС	250000

Найти отдел, работающий над максимальным количеством проектов.

```
SELECT count(project.id_project), department.name
```

```

FROM accounting_for_tasks.department
JOIN accounting_for_tasks.staffing ON staffing.id_department = department.id_department
JOIN accounting_for_tasks.contract ON staffing.id_staffing = contract.id_staffing
JOIN accounting_for_tasks.project ON project.leader = contract.id_contract
group by department.id_department
having count(project.id_project) = (SELECT count(project.id_project)
FROM accounting_for_tasks.department
JOIN accounting_for_tasks.staffing ON staffing.id_department = department.id_department
JOIN accounting_for_tasks.contract ON staffing.id_staffing = contract.id_staffing
JOIN accounting_for_tasks.project ON project.leader = contract.id_contract
group by department.id_department
order by count desc
limit 1)

```

The screenshot shows a SQL IDE interface. The top pane displays a SQL query with 14 lines. The bottom pane shows the results of the query in a table format.

Query:

```

1 SELECT count(project.id_project), department.name
2 FROM accounting_for_tasks.department
3 JOIN accounting_for_tasks.staffing ON staffing.id_department = department.id_department
4 JOIN accounting_for_tasks.contract ON staffing.id_staffing = contract.id_staffing
5 JOIN accounting_for_tasks.project ON project.leader = contract.id_contract
6 group by department.id_department
7 having count(project.id_project) = (SELECT count(project.id_project)
8 FROM accounting_for_tasks.department
9 JOIN accounting_for_tasks.staffing ON staffing.id_department = department.id_department
10 JOIN accounting_for_tasks.contract ON staffing.id_staffing = contract.id_staffing
11 JOIN accounting_for_tasks.project ON project.leader = contract.id_contract
12 group by department.id_department
13 order by count desc
14 limit 1)

```

Data Output:

	count bigint	name character varying (40)
1	2	ДИС

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

```

SELECT employee.fio, project.name as "project", task.description, task.date_end
FROM accounting_for_tasks.employee
LEFT JOIN accounting_for_tasks.contract using(id_employee)
LEFT JOIN accounting_for_tasks.employee_on_task ON employee_on_task.id_contract =
contract.id_contract
LEFT JOIN accounting_for_tasks.task ON employee_on_task.id_task = task.id_task
LEFT JOIN accounting_for_tasks.project ON task.id_project = project.id_project
where (employee.id_employee not in (
select employee.id_employee

```

```

from accounting_for_tasks.project
JOIN accounting_for_tasks.contract ON contract.id_contract = project.leader
JOIN accounting_for_tasks.employee ON contract.id_employee = employee.id_employee
) and task.description is null)
or task.description is not null
union
select employee.fio, project.name, 'lead', project.date_end
from accounting_for_tasks.project
JOIN accounting_for_tasks.contract ON contract.id_contract = project.leader
JOIN accounting_for_tasks.employee ON contract.id_employee = employee.id_employee

```

Query
Query History
Sc

```

1 SELECT employee.fio, project.name as "project", task.description, task.date_end
2 FROM accounting_for_tasks.employee
3 LEFT JOIN accounting_for_tasks.contract using(id_employee)
4 LEFT JOIN accounting_for_tasks.employee_on_task ON employee_on_task.id_contract = contract.id_contract
5 LEFT JOIN accounting_for_tasks.task ON employee_on_task.id_task = task.id_task
6 LEFT JOIN accounting_for_tasks.project ON task.id_project = project.id_project
7 where (employee.id_employee not in (
8     select employee.id_employee
9     from accounting_for_tasks.project
10    JOIN accounting_for_tasks.contract ON contract.id_contract = project.leader
11    JOIN accounting_for_tasks.employee ON contract.id_employee = employee.id_employee
12 ) and task.description is null)
13 or task.description is not null
14 union
15 select employee.fio, project.name, 'lead', project.date_end
16 from accounting_for_tasks.project
17 JOIN accounting_for_tasks.contract ON contract.id_contract = project.leader
18 JOIN accounting_for_tasks.employee ON contract.id_employee = employee.id_employee
19

```

Data Output
Messages
Notifications

	fio character varying (50)	project character varying (50)	description text	date_end date
1	Алексеев Алексей Алексеевич	[null]	[null]	[null]
2	Иванов Иван Иванович	ai project	lead	2024-04-16
3	Сергеев Сергей Сергеевич	my.itmo	lead	2024-07-03
4	Олейникова Полина Леонидовна	my.itmo	страница расписания	2023-08-03
5	Сергеев Сергей Сергеевич	ису	lead	2024-07-01
6	Иванов Иван Иванович	my.itmo	страница персоналий	2023-07-03

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 employee.fio, project.name as "project", task.description as "task",
CURRENT_DATE - task.date_end as "дней просрочено"
FROM accounting_for_tasks.control
JOIN accounting_for_tasks.task using(id_task)
JOIN accounting_for_tasks.employee_on_task using(id_task)
JOIN accounting_for_tasks.contract using(id_contract)
JOIN accounting_for_tasks.staffing using(id_staffing)
JOIN accounting_for_tasks.employee using(id_employee)

```


JOIN accounting_for_tasks.project using(id_project)

where control.status!=100 and task.date_end<Now()

Query Query History

```
1 SELECT employee.fio, project.name as "project", task.description as "task",
2 CURRENT_DATE - task.date_end as "дней просрочено"
3 FROM accounting_for_tasks.control
4 JOIN accounting_for_tasks.task using(id_task)
5 JOIN accounting_for_tasks.employee_on_task using(id_task)
6 JOIN accounting_for_tasks.contract using(id_contract)
7 JOIN accounting_for_tasks.staffing using(id_staffing)
8 JOIN accounting_for_tasks.employee using(id_employee)
9 JOIN accounting_for_tasks.project using(id_project)
10 where control.status!=100 and task.date_end<Now()
11
```

Data Output Messages Notifications

	fio character varying (50)	project character varying (50)	task text	дней просрочено integer
1	Олейникова Полина Леонидовна	my.itmo	страница расписания	14

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

SELECT project.name as "project"

FROM accounting_for_tasks.project

JOIN accounting_for_tasks.task using(id_project)

JOIN accounting_for_tasks.employee_on_task using(id_task)

group by project.id_project

having count(employee_on_task.id_contract)>3

Query Query History

```
1 SELECT project.name as "project"
2 FROM accounting_for_tasks.project
3 JOIN accounting_for_tasks.task using(id_project)
4 JOIN accounting_for_tasks.employee_on_task using(id_task)
5 group by project.id_project
6 having count(employee_on_task.id_contract)>3
7
```

Data Output Messages Notifications

	project character varying (50)
1	my.itmo

1.2 Представления

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

```
CREATE VIEW accounting_for_tasks.info_for_leaders AS
select employee.fio, department.name as "department", project.name as "project", task.description as "task",
task.date_start, task.date_end
from accounting_for_tasks.employee_on_task
join accounting_for_tasks.task using(id_task)
join accounting_for_tasks.project using(id_project)
join accounting_for_tasks.contract using(id_contract)
join accounting_for_tasks.staffing using(id_staffing)
join accounting_for_tasks.employee using(id_employee)
join accounting_for_tasks.department using(id_department)
```

The screenshot shows a query editor with a tab labeled 'Query'. The query text is the same CREATE VIEW statement as above. Below the query, the 'Messages' tab is active, displaying the message: 'CREATE VIEW' and 'Query returned successfully in 77 msec.'

The screenshot shows a query editor with a tab labeled 'Query'. The query text is: `select * from accounting_for_tasks.info_for_leaders`. Below the query, the 'Data Output' tab is active, displaying a table with 7 columns and 4 rows of data.

	fio character varying (50)	department character varying (40)	project character varying (50)	task text	date_start date	date_end date
1	Иванов Иван Иванович	ДИС	my.itmo	страница персоналий	2023-06-03	2023-07-03
2	Олейникова Полина Леонидовна	ДИС	my.itmo	страница расписания	2023-03-03	2023-04-03
3	Сергеев Сергей Сергеевич	ДИС	my.itmo	страница заявок	2023-08-03	2023-09-25
4	Иванов Иван Иванович	ДОД	my.itmo	страница заявок	2023-08-03	2023-09-25

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

```
CREATE VIEW accounting_for_tasks.info_projects AS
select project.name as "project"
from accounting_for_tasks.project
join accounting_for_tasks.task using(id_project)
join accounting_for_tasks.control using(id_task)
where control.status!=100 and project.date_end = current_date
```

```
group by project.id_project  
having count(task.id_task)>1
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 CREATE VIEW accounting_for_tasks.info_projects AS  
2 select project.name as "project"  
3 from accounting_for_tasks.project  
4 join accounting_for_tasks.task using(id_project)  
5 join accounting_for_tasks.control using(id_task)  
6 where control.status!=100 and project.date_end = current_date  
7 group by project.id_project  
8 having count(task.id_task)>1  
9  
10
```

The results pane shows the message: "Query returned successfully in 44 msec."

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 select * from accounting_for_tasks.info_projects
```

The results pane shows the following data:

	project character varying (50)
1	my.itmo

1.3 Модификация данных

Повысить зарплату на 5000 Иванову из ДОДа

```
UPDATE accounting_for_tasks.position  
SET salary = salary + 5000  
WHERE id_position = (SELECT id_position from accounting_for_tasks.contract  
join accounting_for_tasks.staffing using(id_staffing)  
join accounting_for_tasks.employee using(id_employee)  
join accounting_for_tasks.position using(id_position)  
join accounting_for_tasks.department using(id_department)  
WHERE employee.fio='Иванов Иван Иванович' and department.name='ДОД')
```

Query

Query History

1

select * from accounting_for_tasks.position

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑

🗄

⬇

📈

	id_position [PK] integer	name character varying (20)	salary integer
1	2	руководитель	250000
2	1	программист	150000
3	3	лаборант	50000

Query

Query History

1

select * from accounting_for_tasks.position

Data Output

Messages

Notifications

≡+

	id_position [PK] integer	name character varying (20)	salary integer
1	1	программист	150000
2	3	лаборант	50000
3	2	руководитель	255000

Разорвать контракт с Сергеевичем из ДОДа

```
DELETE FROM accounting_for_tasks.contract
WHERE id_contract =
(SELECT id_contract from accounting_for_tasks.contract
join accounting_for_tasks.staffing using(id_staffing)
join accounting_for_tasks.employee using(id_employee)
join accounting_for_tasks.position using(id_position)
join accounting_for_tasks.department using(id_department)
WHERE employee.fio='Сергеев Сергей Сергеевич' and department.name='ДОД')
```

Query

Query History









1 SELECT * FROM accounting_for_tasks.contract




2 ORDER BY id_contract ASC

Data Output

Messages

Notifications



	id_contract [PK] integer 	id_staffing integer 	id_employee integer 
1	5	4	2
2	1	1	1
3	2	3	3
4	3	2	2
5	4	4	1

Query

Query History

1

SELECT * FROM accounting_for_tasks.contract

2

ORDER BY id_contract ASC

Data Output

Messages

Notifications

Добавить лаборанта Иванова на задачу поиска с вознаграждением 7000.

```
INSERT INTO accounting_for_tasks.employee_on_task
```

```
(id_employee_on_task, id_task, id_contract, status, reward)
```

```
VALUES
```

```
( 5,
```

```
(SELECT id_task FROM accounting_for_tasks.task WHERE task.description ='поиск'),
```

```
(SELECT id_contract FROM accounting_for_tasks.contract
```

```
join accounting_for_tasks.staffing using(id_staffing)
```

```

join accounting_for_tasks.position using(id_position)
join accounting_for_tasks.employee using(id_employee)
WHERE employee.fio='Иванов Иван Иванович' and position.name='лаборант'),
0,
7000)

```

Query

Query History









1 SELECT * FROM accounting_for_tasks.employee_on_task

2 ORDER BY id_employee_on_task ASC

Data Output

Messages

Notifications



	id_employee_on_task [PK] integer	id_task integer	id_contract integer	status integer	reward integer
1	1	1	1	1	10000
2	2	2	2	0	6000
3	3	3	3	0	100
4	4	3	4	0	100

Query

Query History









1 SELECT * FROM accounting_for_tasks.employee_on_task

2 ORDER BY id_employee_on_task ASC

Data Output

Messages

Notifications



	id_employee_on_task [PK] integer	id_task integer	id_contract integer	status integer	reward integer
1	1	1	1	1	10000
2	2	2	2	0	6000
3	3	3	3	0	100
4	4	3	4	0	100
5	5	4	1	0	7000

1.4 Создание индексов

План запросов.

1 запрос

Query Query History

```

1 SELECT project.name FROM accounting_for_tasks.project
2 WHERE date_start < (NOW()-INTERVAL '1 month')
3

```

Data Output Messages Explain × Notifications

	name character varying (50)
1	my.itmo

Data Output Messages Explain × Notifications

Graphical Analysis Statistics

project

Data Output Messages Explain × Notifications

Graphical Analysis Statistics

#	Node
1.	→ Seq Scan on project as project Filter: (date_start < (now() - '1 mon::interval))

Data Output Messages Explain × Notifications

Successfully run. Total query runtime: 49 msec.
1 rows affected.

2 запрос

Query Query History

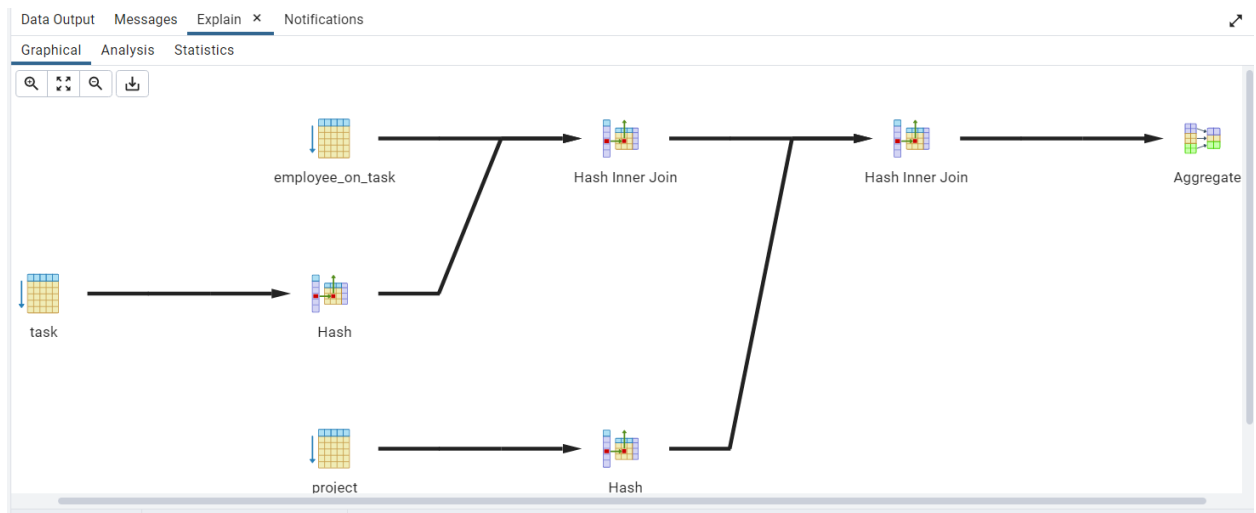
```

1 SELECT project.id_project, project.name as "project"
2 FROM accounting_for_tasks.project
3 JOIN accounting_for_tasks.task using(id_project)
4 JOIN accounting_for_tasks.employee_on_task using(id_task)
5 group by project.id_project
6 having count(employee_on_task.id_contract)>3
7
8

```

Data Output Messages Explain × Notifications

	id_project [PK] integer	project character varying (50)
1	1	my.itmo



#	Node
1.	→ Aggregate Filter: (count(employee_on_task.id_contract) > 3)
2.	→ Hash Inner Join Hash Cond: (task.id_project = project.id_project)
3.	→ Hash Inner Join Hash Cond: (employee_on_task.id_task = task.id_task)
4.	→ Seq Scan on employee_on_task as employee_on_task
5.	→ Hash
6.	→ Seq Scan on task as task
7.	→ Hash
8.	→ Seq Scan on project as project

Data Output Messages Explain × Notifications

Successfully run. Total query runtime: 45 msec.
1 rows affected.

Создание простого индекса для первого запроса:

```
CREATE INDEX i_project ON accounting_for_tasks.project(date_start)
```

Создание составного индекса для второго запроса:

```
CREATE UNIQUE INDEX double_i_task ON accounting_for_tasks.task(id_task, id_project)
```

Проверка созданных индексов:

Query

Query History

Scratch Pad

1

2

```
select * from pg_indexes where tablename='task'
```

Data Output

Messages

Notifications

	<div>schemaname</div> <div>name</div>	<div>tablename</div> <div>name</div>	<div>indexname</div> <div>name</div>	<div>tablespace</div> <div>name</div>	<div>indexdef</div> <div>text</div>
1	accounting_for_tasks	task	task_pkey	[null]	CREATE UNIQUE INDEX task_pkey ON accounting_for_tasks.task USING btree (id_task)
2	accounting_for_tasks	task	fki_id_project	[null]	CREATE INDEX fki_id_project ON accounting_for_tasks.task USING btree (id_project)
3	accounting_for_tasks	task	double_l_task	[null]	CREATE UNIQUE INDEX double_l_task ON accounting_for_tasks.task USING btree (id_task, id_proje...

Query

Query History

Scratch Pad

1

select * from pg_indexes where tablename='project'

2

Data Output

Messages

Notifications

	<div>schemaname</div> <div>name</div>	<div>tablename</div> <div>name</div>	<div>indexname</div> <div>name</div>	<div>tablespace</div> <div>name</div>	<div>indexdef</div> <div>text</div>
1	accounting_for_tasks	project	project_pkey	[null]	CREATE UNIQUE INDEX project_pkey ON accounting_for_tasks.project USING btree (id_project)
2	accounting_for_tasks	project	fki_id_organization	[null]	CREATE INDEX fki_id_organization ON accounting_for_tasks.project USING btree (id_organizati...
3	accounting_for_tasks	project	fki_e	[null]	CREATE INDEX fki_e ON accounting_for_tasks.project USING btree (leader)
4	accounting_for_tasks	project	fki_manager	[null]	CREATE INDEX fki_manager ON accounting_for_tasks.project USING btree (manager)
5	accounting_for_tasks	project	i_project	[null]	CREATE INDEX i_project ON accounting_for_tasks.project USING btree (date_start)

Запросы с индексами

Запрос 1

Data Output	Messages	Explain	Notifications
Graphical	Analysis	Statistics	
#	Node		
1.	→ Seq Scan on project as project Filter: (date_start < (now() - '1 mon'::interval))		

Query	Query History
<pre>1 SELECT project.id_project, project.name FROM accounting_for_tasks.project 2 WHERE date_start < (NOW()-INTERVAL '1 month') 3</pre>	
Data Output	Messages
Successfully run. Total query runtime: 97 msec. 1 rows affected.	

Запрос 2

Data Output	Messages	Explain ×	Notifications
Graphical	Analysis	Statistics	
#	Node		
1.	→ Aggregate Filter: (count(employee_on_task.id_contract) > 3)		
2.	→ Hash Inner Join Hash Cond: (employee_on_task.id_task = task.id_task)		
3.	→ Seq Scan on employee_on_task as employee_on_task		
4.	→ Hash		
5.	→ Hash Inner Join Hash Cond: (task.id_project = project.id_project)		
6.	→ Seq Scan on task as task		
7.	→ Hash		
8.	→ Seq Scan on project as project		

Data Output	Messages	Explain ×	Notifications
Successfully run. Total query runtime: 74 msec. 1 rows affected.			

Удаление индексов

```
drop index accounting_for_tasks.double_i_task  
drop index accounting_for_tasks.i_project
```

ЗАКЛЮЧЕНИЕ

SQL запросы позволяют изменять, добавлять или удалять данные, а также составлять различные выборки, подсчитывать числовые характеристики. Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись примерно столько же. Это связано с небольшим количеством данных в таблице.