

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»  
Факультет инфокоммуникационных технологий

**ОТЧЕТ**  
**О ЛАБОРАТОРНОЙ РАБОТЕ № 4**  
по теме: Запросы на выборку данных к БД PostgreSQL.  
по дисциплине: Проектирование и реализация баз данных

**Специальность:**  
09.03.03 Мобильные и сетевые технологии

**Проверил:**  
Говорова М.М. \_\_\_\_\_  
**Дата:** «\_\_»\_\_\_\_\_ 2021 г.  
**Оценка** \_\_\_\_\_

**Выполнил:**  
студентка группы К3240  
Балдина Д.Д.

Санкт-Петербург 2022 г.

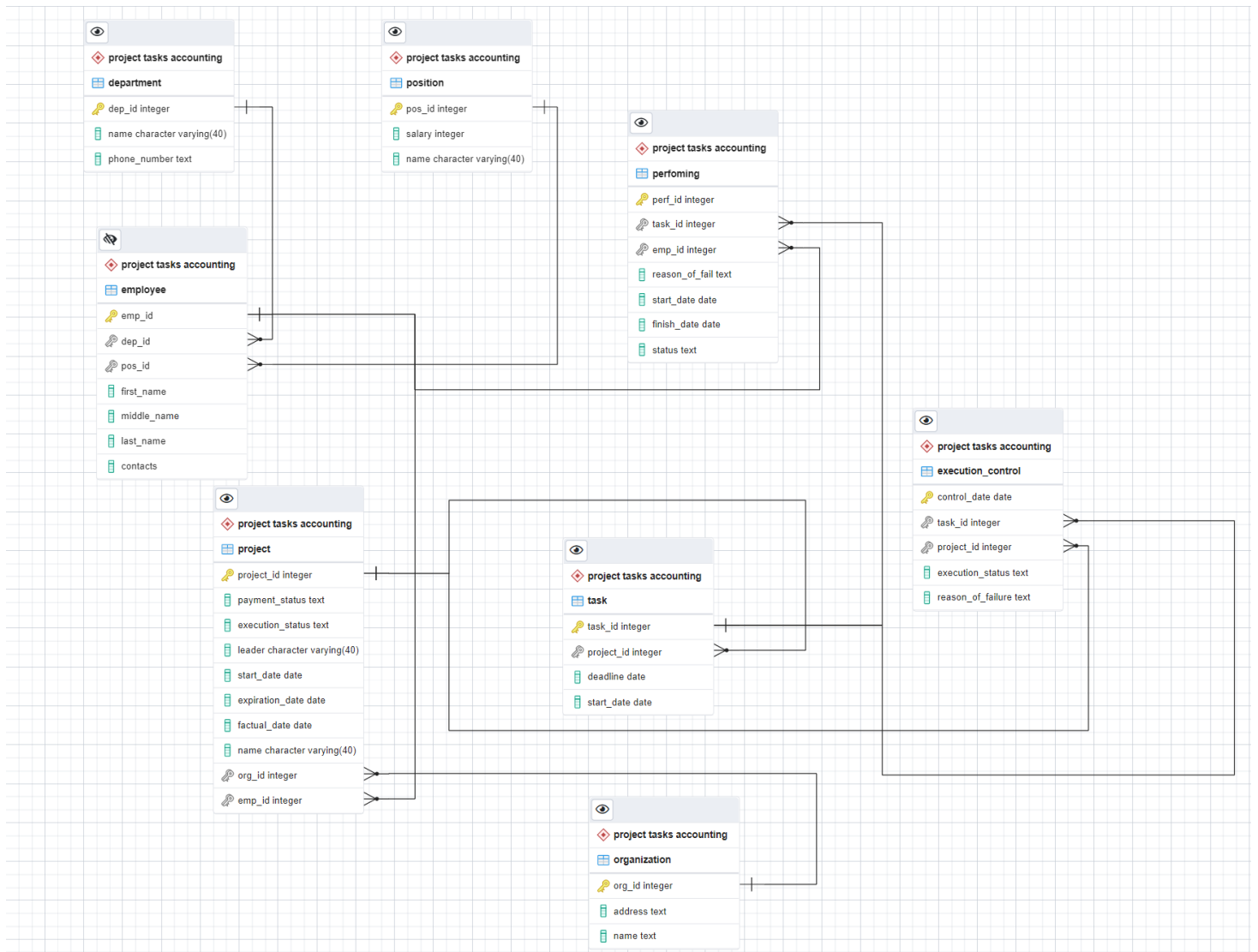
## **ЦЕЛЬ РАБОТЫ**

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL и использования подзапросов при модификации данных.

## **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 посмотреть 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**



## Задание 2. Запросы

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

**Запрос:** SELECT task.task\_id as "Задание", project.name as "Проект", organization.name as "Организация", department.name as "Отдел", CONCAT(employee.first\_name, employee.last\_name) as "Исполнитель"

FROM ps\_tasks.task

INNER JOIN ps\_tasks.project ON project.project\_id = task.project\_id

INNER JOIN ps\_tasks.employee ON project.project\_id = employee.project\_id

INNER JOIN ps\_tasks.organization ON project.org\_id = organization.org\_id

INNER JOIN ps\_tasks.department ON employee.dep\_id = department.dep\_id

ORDER BY task.task\_id

postgres/postgres@PostgreSQL 14					
Query Editor История запросов					
<pre>1 SELECT task.task_id as "Задание", project.name as "Проект", organization.name as "Организация", department.name as "Отдел", 2 CONCAT(employee.first_name, ' ', employee.last_name) as "Исполнитель" 3 FROM ps_tasks.task 4 INNER JOIN ps_tasks.project ON project.project_id = task.project_id 5 INNER JOIN ps_tasks.employee ON project.project_id = employee.project_id 6 INNER JOIN ps_tasks.organization ON project.org_id = organization.org_id 7 INNER JOIN ps_tasks.department ON employee.dep_id = department.dep_id 8 ORDER BY task.task_id 9</pre>					
Результат План выполнения Сообщения Notifications					
Задание integer	Проект character varying (40)	Организация text	Отдел character varying (40)	Исполнитель text	
13	4 Project "Leader"	Adventure	General	Polina Mat	
14	4 Project "Leader"	Adventure	Analytics	Anna Kobel	
15	5 Project "Leader"	Adventure	Fanance	Daria Bor	
16	5 Project "Leader"	Adventure	Fanance	Lev Askin	
17	5 Project "Leader"	Adventure	General	Daria	
18	5 Project "Leader"	Adventure	Analytics	Anna Kobel	
19	5 Project "Leader"	Adventure	General	Polina Mat	
20	6 Project "Leader"	Adventure	Analytics	Anna Kobel	
21	6 Project "Leader"	Adventure	Fanance	Lev Askin	
22	6 Project "Leader"	Adventure	Fanance	Daria Bor	
23	6 Project "Leader"	Adventure	General	Daria	
24	6 Project "Leader"	Adventure	General	Polina Mat	

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

**Запрос:** SELECT project.project\_id, project.name FROM ps\_tasks.project  
WHERE start\_date < (NOW()-INTERVAL '1 month')

```
1 SELECT project.project_id, project.name FROM ps_tasks.project
2 WHERE start_date < (NOW()-INTERVAL '1 month')
```

Результат			
План выполнения			
Сообщения			
Notifications			
	project_id [PK] integer	name character varying (40)	
1	2	Project "Leader"	
2	3	"Finance system"	
3	1	Create the game "Future"	

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

**Запрос:** SELECT emp\_id, CONCAT(employee.first\_name, ' ',  
employee.middle\_name, ' ', employee.last\_name) as "Исполнитель",  
department.name as "Отдел", employee.salary as "Зарплата"  
FROM ps\_tasks.employee  
JOIN ps\_tasks.department ON employee.dep\_id = department.dep\_id  
WHERE employee.salary > (  
SELECT AVG(salary)  
FROM ps\_tasks.employee  
JOIN ps\_tasks.department ON employee.dep\_id = department.dep\_id  
)  
ORDER BY department.name

```

1 SELECT emp_id, CONCAT(employee.first_name, ' ', employee.middle_name, ' ', employee.last_name) as "Исполнитель",
2 department.name as "Отдел", employee.salary as "Зарплата"
3 FROM ps_tasks.employee
4 JOIN ps_tasks.department ON employee.dep_id = department.dep_id
5 WHERE employee.salary > (
6     SELECT AVG(salary)
7     FROM ps_tasks.employee
8     JOIN ps_tasks.department ON employee.dep_id = department.dep_id
9 )
10 ORDER BY department.name
11
12

```

Результат	План выполнения	Сообщения	Notifications
emp_id integer	Исполнитель text	Отдел character varying (40)	Зарплата double precision
1	5 Anna Ivanovna Kobel	Analytics	100000
2	7 Zak Newman	Analytics	140000
3	6 Lev Victorovich Askin	Fanance	110000
4	8 Daria Danilovna Bor	Fanance	98800
5	2 Olga Danilovna Zilina	Fanance	120000
6	9 Polina Alekseevna Mat	General	95600

4. Найти отдел, работающий над максимальным количеством проектов.

**Запрос:**

```

SELECT "Отдел"
FROM
(SELECT employee.dep_id, COUNT(distinct employee.project_id), department.name as
"Отдел"
FROM ps_tasks.employee
JOIN ps_tasks.department on employee.dep_id = department.dep_id
WHERE project_id IS NOT NULL
GROUP BY employee.dep_id, department.name
ORDER BY COUNT(distinct employee.project_id) desc) as count_table

```

```

WHERE count = (SELECT MAX(count)
FROM
(SELECT employee.dep_id, COUNT(distinct employee.project_id), department.name as
"Department"
FROM ps_tasks.employee
JOIN ps_tasks.department on employee.dep_id = department.dep_id
WHERE project_id IS NOT NULL
GROUP BY employee.dep_id, department.name
ORDER BY COUNT(distinct employee.project_id) desc) as count_table)

```

```

1 SELECT "Отдел"
2 FROM
3 (SELECT employee.dep_id, COUNT(distinct employee.project_id), department.name as "Отдел"
4 FROM ps_tasks.employee
5 JOIN ps_tasks.department on employee.dep_id = department.dep_id
6 WHERE project_id IS NOT NULL
7 GROUP BY employee.dep_id, department.name
8 ORDER BY COUNT(distinct employee.project_id) desc) as count_table
9
10 WHERE count = (SELECT MAX(count)
11 FROM |
12 (SELECT employee.dep_id, COUNT(distinct employee.project_id), department.name as "Department"
13 FROM ps_tasks.employee
14 JOIN ps_tasks.department on employee.dep_id = department.dep_id
15 WHERE project_id IS NOT NULL
16 GROUP BY employee.dep_id, department.name
17 ORDER BY COUNT(distinct employee.project_id) desc) as count_table)

```

Результат План выполнения Сообщения Notifications

	Отдел character varying (40) 🔒
1	Analytics

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

**Запрос:** SELECT CONCAT(employee.first\_name, employee.last\_name) as "Исполнитель", project.name as "Проект", task.task\_id as "Задание", task.deadline as "Дата выполнения задания"

FROM ps\_tasks.task

INNER JOIN ps\_tasks.project ON project.project\_id = task.project\_id

RIGHT JOIN ps\_tasks.employee ON project.project\_id = employee.project\_id

INNER JOIN ps\_tasks.department ON employee.dep\_id = department.dep\_id

ORDER BY task.task\_id

```

1 SELECT CONCAT(employee.first_name, employee.last_name) as "Исполнитель", project.name as "Проект", task.task_id as "Задание",
2 task.deadline as "Дата выполнения задания"
3 FROM ps_tasks.task
4 INNER JOIN ps_tasks.project ON project.project_id = task.project_id
5 RIGHT JOIN ps_tasks.employee ON project.project_id = employee.project_id
6 INNER JOIN ps_tasks.department ON employee.dep_id = department.dep_id
7 ORDER BY task.task_id

```

Результат План выполнения Сообщения Notifications

	Исполнитель text	Проект character varying (40)	Задание integer	Дата выполнения задания date
14	PolinaMat	Project "Leader"	4	2022-03-24
15	DariaBor	Project "Leader"	5	2022-03-24
16	PolinaMat	Project "Leader"	5	2022-03-24
17	Daria	Project "Leader"	5	2022-03-24
18	LevAskin	Project "Leader"	5	2022-03-24
19	AnnaKobel	Project "Leader"	5	2022-03-24
20	PolinaMat	Project "Leader"	6	2022-03-24
21	DariaBor	Project "Leader"	6	2022-03-24
22	LevAskin	Project "Leader"	6	2022-03-24
23	AnnaKobel	Project "Leader"	6	2022-03-24
24	Daria	Project "Leader"	6	2022-03-24
25	NataliaKochkina	[null]	[null]	[null]

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

**Запрос:** SELECT CONCAT(employee.first\_name, ' ', employee.last\_name) as "Исполнитель", task.task\_id as "Задание",  
CONCAT(NOW() - task.deadline) as "Просроченные дни"  
FROM ps\_tasks.task  
INNER JOIN ps\_tasks.project ON project.project\_id = task.project\_id  
INNER JOIN ps\_tasks.employee ON project.project\_id = employee.project\_id  
WHERE task.task\_status IN ('In work', 'in work', 'Suspended', 'suspended') AND task.deadline < NOW()  
AND project.execution\_status NOT IN ('Finished', 'finished', 'Canceled', 'canceled')



```

1 SELECT CONCAT(employee.first_name, ' ', employee.last_name) as "Исполнитель", task.task_id as "Задание",
2 CONCAT(NOW() - task.deadline) as "Просроченные дни"
3 FROM ps_tasks.task
4 INNER JOIN ps_tasks.project ON project.project_id = task.project_id
5 INNER JOIN ps_tasks.employee ON project.project_id = employee.project_id
6 WHERE task.task_status IN ('In work', 'in work', 'Suspended', 'suspended') AND task.deadline < NOW()
7 AND project.execution_status NOT IN ('Finished', 'finished', 'Canceled', 'canceled')

```

Результат    План выполнения    Сообщения    Notifications

	Исполнитель text	Задание integer	Просроченные дни text	
1	Ivan Ivanov	1	9 days 19:55:22.030097	
2	Andrew Shepel	1	9 days 19:55:22.030097	
3	Olga Zilina	1	9 days 19:55:22.030097	

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

**Запрос:** SELECT project.project\_id, project.name as "Проект"

FROM ps\_tasks.project

JOIN ps\_tasks.employee ON project.project\_id = employee.project\_id

GROUP BY project.name, project.project\_id

HAVING COUNT(employee.project\_id) > 3

```

1 SELECT project.project_id, project.name as "Проект"
2 FROM ps_tasks.project
3 JOIN ps_tasks.employee ON project.project_id = employee.project_id
4 GROUP BY project.name, project.project_id
5 HAVING COUNT(employee.project_id) > 3

```

Результат    План выполнения    Сообщения    Notifications

	project_id [PK] integer	Проект character varying (40)	
1	1	Create the game "Future"	
2	2	Project "Leader"	

### Задание 3. Представления

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

```

CREATE VIEW ps_tasks.info_for_leaders AS
SELECT department.name as "Отдел", CONCAT(employee.first_name,
employee.last_name) as "Исполнитель", project.name as "Проект",
task.task_id as "Задание", task.deadline as "Срок выполнения задания"
FROM ps_tasks.task
INNER JOIN ps_tasks.project ON project.project_id = task.project_id
INNER JOIN ps_tasks.employee ON project.project_id = employee.project_id
INNER JOIN ps_tasks.department ON employee.dep_id = department.dep_id
ORDER BY task.task_id

```

```

1 CREATE VIEW ps_tasks.info_for_leaders AS
2 SELECT department.name as "Отдел", CONCAT(employee.first_name, employee.last_name) as "Исполнитель", project.name as "Проект",
3 task.task_id as "Задание", task.deadline as "Срок выполнения задания"
4 FROM ps_tasks.task
5 INNER JOIN ps_tasks.project ON project.project_id = task.project_id
6 INNER JOIN ps_tasks.employee ON project.project_id = employee.project_id
7 INNER JOIN ps_tasks.department ON employee.dep_id = department.dep_id
8 ORDER BY task.task_id

```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 105 msec.

```

1 SELECT * FROM ps_tasks.info_for_leaders

```

Результат План выполнения Сообщения Notifications

	Отдел character varying (40)	Исполнитель text	Проект character varying (40)	Задание integer	Срок выполнения задания date
10	Fanance	DariaBor	Project "Leader"	4	2022-03-24
11	Analytics	AnnaKobel	Project "Leader"	4	2022-03-24
12	General	Daria	Project "Leader"	4	2022-03-24
13	Fanance	LevAskin	Project "Leader"	4	2022-03-24
14	Fanance	LevAskin	Project "Leader"	5	2022-03-24
15	Analytics	AnnaKobel	Project "Leader"	5	2022-03-24
16	General	Daria	Project "Leader"	5	2022-03-24
17	Fanance	DariaBor	Project "Leader"	5	2022-03-24
18	Fanance	LevAskin	Project "Leader"	6	2022-03-24
19	Analytics	AnnaKobel	Project "Leader"	6	2022-03-24
20	General	Daria	Project "Leader"	6	2022-03-24
21	Fanance	DariaBor	Project "Leader"	6	2022-03-24

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

```
CREATE VIEW ps_tasks.project_info AS
SELECT project.project_id, project.name as "Проект", COUNT(task.task_id)
FROM ps_tasks.project
JOIN ps_tasks.task ON task.project_id = project.project_id
WHERE project.execution_status IN ('In work','in work') AND project.expiration_date
= NOW()::date
GROUP BY project.project_id, project.name
HAVING COUNT(task.task_id) > 3
```

```
1 CREATE VIEW ps_tasks.project_info AS
2 SELECT project.project_id, project.name as "Проект", COUNT(task.task_id)
3 FROM ps_tasks.project
4 JOIN ps_tasks.task ON task.project_id = project.project_id
5 WHERE project.execution_status IN ('In work','in work') AND project.expiration_date = NOW()::date
6 GROUP BY project.project_id, project.name
7 HAVING COUNT(task.task_id) > 3
```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 35 msec.

```
1 SELECT * FROM ps_tasks.project_info
```

Результат План выполнения Сообщения Notifications

	project_id integer	Проект character varying (40)	count bigint	
1	4	Mobile device	4	

## Задание 4. Модификация данных

### 1. Запрос с UPDATE

Повысить зарплату на 5000 всем, у кого она меньше 80000

До:

1 SELECT \* FROM ps\_tasks.employee

Результат

План выполнения

Сообщения

Notifications

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	1	111	1	Ivan	Olegovich	Ivanov	iiii@mail.ru	1	75000
3	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
4	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
5	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
6	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
7	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
8	10	111	3	Natalia	[null]	Kochkina	koch@mail.ru	[null]	20000
9	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
10	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600

```
UPDATE ps_tasks.employee
SET salary = salary + 5000
WHERE salary IN (
SELECT salary FROM ps_tasks.employee
WHERE salary < 80000)
```

```
1 UPDATE ps_tasks.employee
2 SET salary = salary + 5000
3 WHERE salary IN (
4     SELECT salary FROM ps_tasks.employee
5     WHERE salary < 80000)
```

После:

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
6	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9	1	111	1	Ivan	Olegovich	Ivanov	iiiiv@mail.ru	1	80000
10	10	111	3	Natalia	[null]	Kochkina	koch@mail.ru	[null]	25000

## 2. Запрос с DELETE

Удалить работника, который получает меньше 40000

До:

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
6	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9	1	111	1	Ivan	Olegovich	Ivanov	iiiiv@mail.ru	1	80000
10	10	111	3	Natalia	[null]	Kochkina	koch@mail.ru	[null]	25000

```
DELETE FROM ps_tasks.employee
WHERE project_id IN (
    SELECT project_id FROM ps_tasks.employee
    WHERE project_id IS NULL
)
```

```
DELETE FROM ps_tasks.employee
WHERE salary IN (
    SELECT salary FROM ps_tasks.employee
    WHERE salary < 40000
)
```

После:

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
6	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9	1	111	1	Ivan	Olegovich	Ivanov	iiiiv@mail.ru	1	80000

### 3. Запрос с INSERT

Добавление новых сотрудников

До:

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1	3	333	3	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2	4	333	3	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3	5	111	2	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4	6	222	3	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5	7	111	1	Zak	[null]	Newman	zakn@gmail.com	3	140000
6	8	222	2	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7	2	222	2	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8	9	333	1	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9	1	111	1	Ivan	Olegovich	Ivanov	iiiiv@mail.ru	1	80000

```
INSERT INTO ps_tasks.employee (emp_id, dep_id, pos_id, first_name, middle_name,
last_name, contacts, project_id, salary)
```

```
VALUES
```

```
( 10,
  333,
  2,
  'Mark',
  'Davidivich',
  'Morsov',
  'mor@mail.ru',
```

```
(SELECT project_id FROM ps_tasks.project WHERE project.name ='Create the game
"Future"),
75000
)
```

```
INSERT INTO ps_tasks.employee (emp_id, dep_id, pos_id, first_name, middle_name, last_name, contacts, project_id, salary)
VALUES
(
10,
333,
2,
'Mark',
'Davidivich',
'Morsov',
'mor@mail.ru',
(SELECT project_id FROM ps_tasks.project WHERE project.name ='Create the game "Future"'),
75000
)
```

После:

	emp_id [PK] integer	dep_id integer	pos_id integer	first_name text	middle_name text	last_name text	contacts text	project_id integer	salary double precision
1		3	333	Daria	[null]	Krasnova	dar@mail.ru, +283285	2	80000
2		4	333	Andrew	[null]	Shepel	sh@amail.ru	1	90000
3		5	111	Anna	Ivanovna	Kobel	ko@gmail.ru	2	100000
4		6	222	Lev	Victorovich	Askin	levv@mail.ru	2	110000
5		7	111	Zak	[null]	Newman	zakn@gmail.com	3	140000
6		8	222	Daria	Danilovna	Bor	borda@mail.ru	2	98800
7		2	222	Olga	Danilovna	Zilina	zhil@gmail.com	1	120000
8		9	333	Polina	Alekseevna	Mat	pol@mail.ru	1	95600
9		1	111	Ivan	Olegovich	Ivanov	iiii@mail.ru	1	80000
10		10	333	Mark	Davidivich	Morsov	mor@mail.ru	1	75000

## Задание 5. История запросов

Query Editor
История запросов

Show queries generated internally by pgAdmin?
Да

```

▶ SELECT employee.dep_id, department.name as "Отдел", COUNT(e...
17:37:53
▶ SELECT employee.dep_id, department.name as "Отдел", COUNT(e...
17:37:57
▶ SELECT department.dep_id, department.name as "Отдел", COUNT...
17:31:57
▶ SELECT department.dep_id, department.name as "Отдел", COUNT...
17:31:23
▶ SELECT department.dep_id, department.name as "Отдел", COUNT...
17:31:82
▶ SELECT department.dep_id, department.name as "Отдел", COUNT...
17:30:53
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mi...
17:27:18
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.la...
17:27:68
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.la...
17:29:19
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mi...
17:24:45
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mi...
17:23:21
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mid...
17:21:40
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mid...
17:20:53
▶ SELECT emp_id, CONCAT(employee.first_name, ' ', employee.mid...
17:18:13

```

24.03.2022 17:52:43
3
35 msec

Дата
Rows Affected
Продолжительность

Copy
Copy to Query Editor

```

SELECT employee.dep_id, COUNT(employee.dep_id)
FROM ps_tasks.employee
GROUP BY employee.dep_id

```

Сообщения
Запрос выполнен успешно. Общее время выполнения: 35 msec.  
обработано строк: 3.



# Задание 6. EXPLAIN

## 1. Без индексов

### Запрос №2

План запроса:

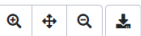
```
1 EXPLAIN SELECT project.project_id, project.name FROM ps_tasks.project
2 WHERE start_date < (NOW()-INTERVAL '1 month')
3
```

Результат    План выполнения    Сообщения    Notifications

	QUERY PLAN	
	text	
1	Seq Scan on project (cost=0.00..1.05 rows=1 width=102)	
2	[...] Filter: (start_date < (now() - '1 mon':interval))	

Графическое выполнение запроса:

Graphical    Analysis    Статистика



project

project		×
Node Type	Seq Scan	
Parallel Aware	false	
Async Capable	false	
Relation Name	project	
Alias	project	
Actual Rows	4	
Actual Loops	1	
Filter	(start_date < (now() - '1 mon':interval))	
Rows Removed by Filter	0	
loops	1	
_serial	1	

Время выполнения запроса:

Запрос выполнен успешно. Общее время выполнения: 39 msec.  
обработано строк: 1.

### Запрос №7

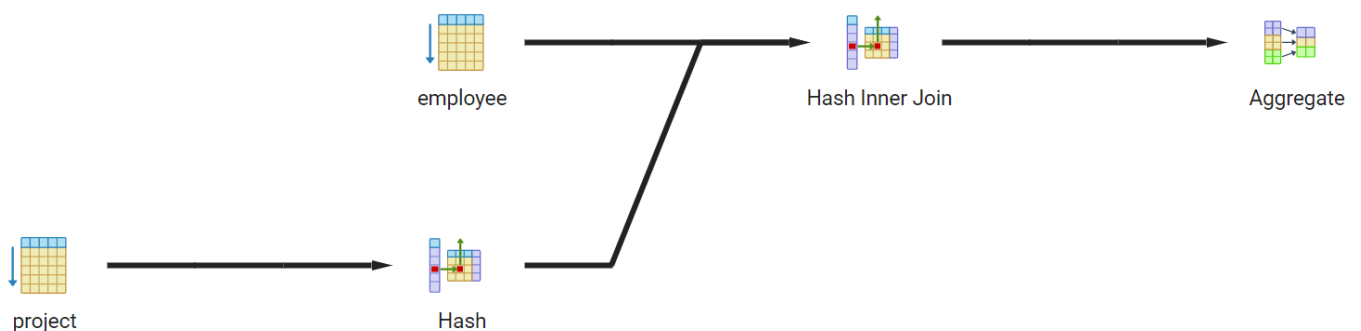
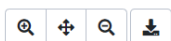
## План запроса:

```
1 EXPLAIN SELECT project.project_id, project.name as "Проект"
2 FROM ps_tasks.project
3 JOIN ps_tasks.employee ON project.project_id = employee.project_id
4 GROUP BY project.name, project.project_id
5 HAVING COUNT(employee.project_id) > 3
6
7
```

Результат План выполнения Сообщения Notifications

	QUERY PLAN	
	text	🔒
1	HashAggregate (cost=2.25..2.29 rows=1 width=102)	
2	[...] Group Key: project.project_id	
3	[...] Filter: (count(employee.project_id) > 3)	
4	[...] -> Hash Join (cost=1.07..2.20 rows=10 width=106)	
5	[...] Hash Cond: (employee.project_id = project.project_id)	
6	[...] -> Seq Scan on employee (cost=0.00..1.10 rows=10 width=4)	
7	[...] -> Hash (cost=1.03..1.03 rows=3 width=102)	
8	[...] -> Seq Scan on project (cost=0.00..1.03 rows=3 width=102)	

## Графическое представление:



#	Node	Rows	
		Actual	Loops
1.	→ Aggregate (rows=2 loops=1) Filter: (count(employee.project_id) > 3) Rows Removed by Filter: 1 Buckets: Batches: Memory Usage: 24 kB		2
2.	→ Hash Inner Join (rows=10 loops=1) Hash Cond: (employee.project_id = project.project_id)	10	1
3.	→ Seq Scan on employee as employee (rows=10 loops=1)	10	1
4.	→ Hash (rows=4 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	4	1
5.	→ Seq Scan on project as project (rows=4 loops=1)	4	1

Время выполнения запроса:

Запрос выполнен успешно. Общее время выполнения: 36 msec.  
обработано строк: 1.

## 2. Создание индексов

Создание простого индекса для первого(запрос№2) запроса:

CREATE INDEX i\_prj ON ps\_tasks.project(project\_id)

```
1 CREATE INDEX i_prj ON ps_tasks.project(project_id)
```

Создание составного индекса для второго(запрос№7) запроса:

CREATE UNIQUE INDEX double\_index ON ps\_tasks.project(project\_id, name)

```
1 CREATE UNIQUE INDEX double_index ON ps_tasks.project(project_id, name)
```

## Проверка созданных индексов:

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	ps_tasks	project	project_pkey	[null]	CREATE UNIQUE INDEX project_pkey ON ps_tasks.project USING btree (project_id)
2	ps_tasks	project	i_prj	[null]	CREATE INDEX i_prj ON ps_tasks.project USING btree (project_id)
3	ps_tasks	project	double_index	[null]	CREATE UNIQUE INDEX double_index ON ps_tasks.project USING btree (project_id, ...)

## 3. Запросы с индексами

Запрос №2:

План запроса:


	QUERY PLAN text
1	Seq Scan on project (cost=0.00..1.07 rows=1 width=102)
2	[...] Filter: (start_date < (now() - '1 mon'::interval))

Время выполнения запроса:

Запрос выполнен успешно. Общее время выполнения: 56 msec.  
обработано строк: 2.

### Запрос №7:

План запроса:

	QUERY PLAN	
	text	
1	HashAggregate (cost=2.28..2.33 rows=1 width=102)	
2	[...] Group Key: project.project_id	
3	[...] Filter: (count(employee.project_id) > 3)	
4	[...] -> Hash Join (cost=1.09..2.23 rows=10 width=106)	
5	[...] Hash Cond: (employee.project_id = project.project_id)	
6	[...] -> Seq Scan on employee (cost=0.00..1.10 rows=10 width=4)	
7	[...] -> Hash (cost=1.04..1.04 rows=4 width=102)	
8	[...] -> Seq Scan on project (cost=0.00..1.04 rows=4 width=102)	

Время выполнения запроса:

Запрос выполнен успешно. Общее время выполнения: 47 msec.  
обработано строк: 8.

### Задание 7. Удаление индексов:

```
1 DROP INDEX ps_tasks.i_prj
```

Результат    План выполнения    Сообщения    Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 34 msec.

```
1 DROP INDEX ps_tasks.double_index
```




Результат   План выполнения   Сообщения   Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 43 msec.

```
1 SELECT * FROM pg_indexes WHERE tablename = 'project'
```

Результат   План выполнения   Сообщения   Notifications

	 schemaname name	 tablename name	 indexname name	 tablespace name	 indexdef text	
1	ps_tasks	project	project_pkey	[null]	CREATE UNIQUE INDEX project_pkey ON ps_tasks.project USING btree (project_id)	

## ВЫВОДЫ

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 немного медленнее. Это связано с небольшим количеством данных в таблице.