

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет
По Лабораторной работе 2
Создание БД в СУБД PostgreSQL. Резервное копирование и восстановление
БД
по дисциплине **«Базы данных»**

Автор: Горбатов Дмитрий Алексеевич

Факультет: ИКТ

Группа: K32402

Преподаватель: Говорова Марина Михайловна

Санкт-Петербург 2023

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

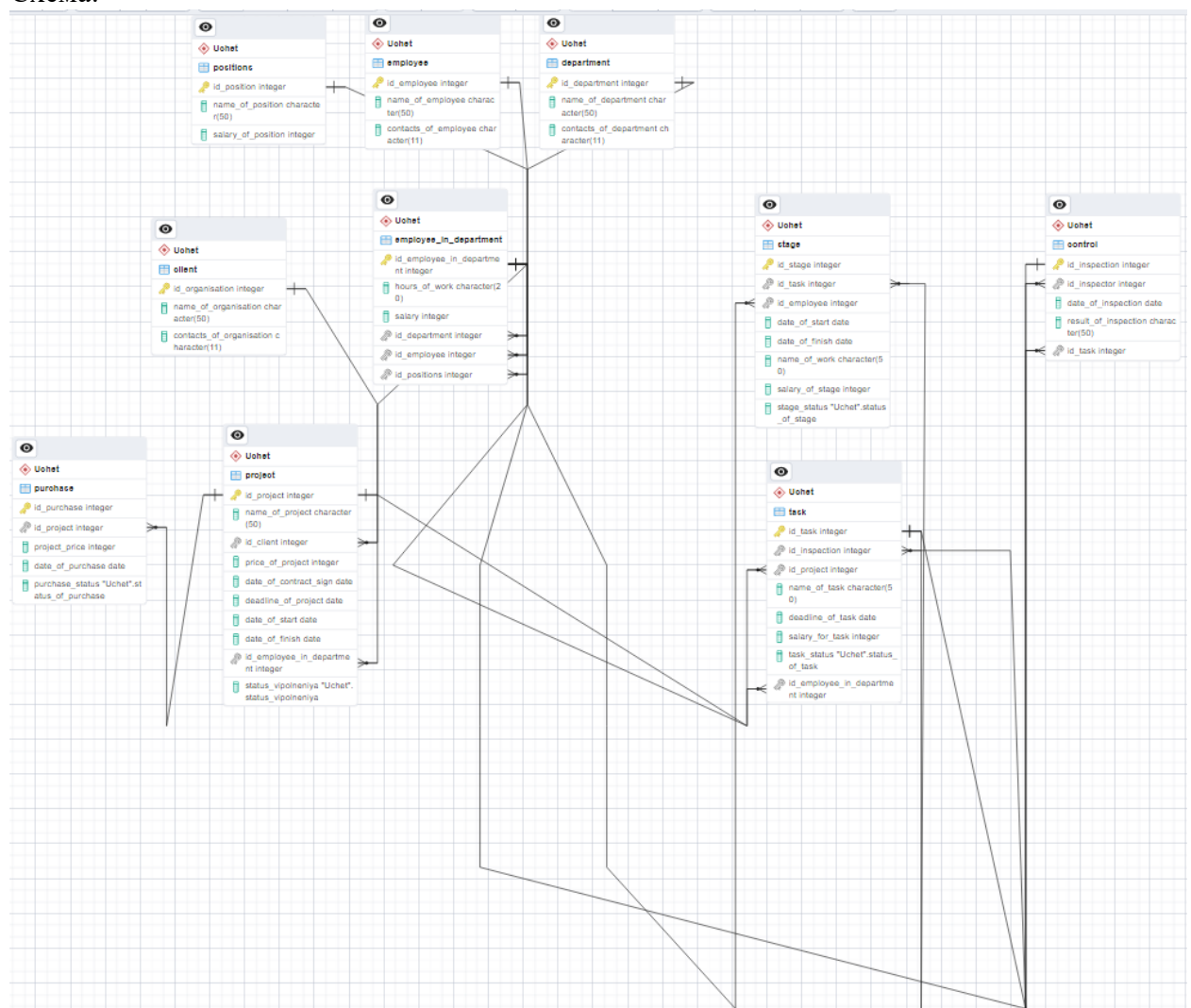
Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

Практическое задание:

1. Создать запросы 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:



Задание 2. Создайте запросы:

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

Код запроса:

```
select task.id_task, project.id_project, client.id_organisation,  
employee.name_of_employee, department.name_of_department
```

```
from "Uchet".task, "Uchet".project, "Uchet".client, "Uchet".employee,  
"Uchet".department, "Uchet".stage, "Uchet".employee_in_department
```

```
where task.id_project=project.id_project and client.id_organisation=project.id_client  
and task.id_task=stage.id_task and  
stage.id_employee=employee_in_department.id_employee_in_department and  
employee_in_department.id_employee=employee.id_employee and  
employee_in_department.id_department=department.id_department
```

order by task.id_task

Результат запроса:

	id_task integer	id_project integer	id_organisation integer	name_of_employee character (50)	name_of_department character (50)
1	110	1	1	LENA	OPERATION
2	111	1	1	LENA	OPERATION
3	111	1	1	VLADISLAV	OPERATION
4	111	1	1	VLADISLAV	OPERATION
5	112	3	3	VLADISLAV	OPERATION
6	112	3	3	VLADISLAV	OPERATION
7	112	3	3	VLADISLAV	OPERATION
8	113	3	3	LENA	OPERATION
9	113	3	3	LENA	OPERATION
10	113	3	3	LENA	OPERATION

Total rows: 10 of 10 Query complete 00:00:01.417

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

Код запроса:

```
select id_project, name_of_project, date_of_start from "Uchet".project where current_date -  
date_of_start > 30 order by id_project
```

Результат запроса:

	id_project [PK] integer	name_of_project character (50)	date_of_start date
1	1	TEST_1	2023-04-15
2	3	HW_2	2023-04-14

Total rows: 2 of 2 Query complete 00:00:00.361

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

Код запроса:

FROM "Uchet".employee_in_department

WHERE salary > (SELECT AVG(salary) FROM "Uchet".employee_in_department);

Результат запроса:

	id_employee_in_department [PK] integer	hours_of_work character (20)	salary integer	id_department integer	id_employee integer	id_positions integer
1	13	36	40000	2	345	1

Total rows: 1 of 1 Query complete 00:00:00.722

4. Найти отдел, работающий над максимальным количеством проектов.

Код запроса:

SELECT id_employee_in_department, COUNT(*) as num_projects

FROM "Uchet".project

GROUP BY id_employee_in_department

ORDER BY num_projects DESC

Результат запроса:

	id_employee_in_department integer	num_projects bigint
1	14	2
2	13	1

Total rows: 2 of 2 Query complete 00:00:00.416

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

Код запроса:

SELECT

employee.name_of_employee AS employee_name,

project.name_of_project AS project_name,

task.name_of_task AS task_name,

task.deadline_of_task AS task_deadline

FROM employee

LEFT JOIN employee_in_department ON employee.id_employee =
employee_in_department.id_employee

LEFT JOIN project ON employee_in_department.id_employee =
project.id_employee_in_department

LEFT JOIN task ON project.id_project = task.id_project

ORDER BY employee_name ASC;

Результат запроса:





	employee_name character (50)	project_name character (50)	task_name character (50)	task_deadline date
1	ALFRED	[null]	[null]	[null]
2	DIMA	[null]	[null]	[null]
3	FEDOR	[null]	[null]	[null]
4	GEORGE	[null]	[null]	[null]
5	GOSHA	[null]	[null]	[null]
6	LENA	TEST_1	сбор материала	2023-04-19
7	LENA	TEST_1	выполнение	2023-04-25
8	LENA	HW_1	сбор материала	2023-05-14
9	VITYA	[null]	[null]	[null]
10	VLADISLAV	HW_2	сбор материала	2023-04-17
11	VLADISLAV	HW_2	выполнение	2023-04-23
Total rows: 11 of 11 Query complete 00:00:00.320				

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.

Код запроса:

```
select employee.name_of_employee, project.name_of_project,
task.id_task, (current_date - task.deadline_of_task) as days_overdue
from "Uchet".employee, "Uchet".project, "Uchet".task, "Uchet".employee_in_department
where task.task_status not in ('Завершен') and task.deadline_of_task < current_date and
employee.id_employee = employee_in_department.id_employee and
employee_in_department.id_employee_in_department = task.id_employee_in_department
and project.id_project = task.id_project
```


Результат запроса:

	name_of_employee character (50) 	name_of_project character (50) 	id_task integer 	days_overdue integer 
1	VLADISLAV	HW_1	109	18
2	VLADISLAV	TEST_1	110	37

Total rows: 2 of 2 Query complete 00:00:00.239

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

Код запроса:

name_of_project
character (50) 

Total rows: 0 of 0 Query complete 00:00:00.358

Задание 2. Создать представления:

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

Код запроса:

```
create view "Uchet".management as select employee.name_of_employee,  
  
department.name_of_department, project.name_of_project,  
  
task.id_task, task.deadline_of_task  
  
from "Uchet".employee, "Uchet".department, "Uchet".project,  
  
"Uchet".task, "Uchet".employee_in_department  
  
where employee.id_employee = employee_in_department.id_employee and  
department.id_department = employee_in_department.id_department  
  
and employee_in_department.id_employee_in_department =  
task.id_employee_in_department and task.id_project = project.id_project
```

Результат запроса:

1

```
select * from "Uchet".management
```

Data Output

Messages

Notifications

	name_of_employee character (50)	name_of_department character (50)	name_of_project character (50)	id_task integer	deadline_of_task date
1	VLADISLAV	OPERATION	HW_1	109	2023-05-14
2	VLADISLAV	OPERATION	TEST_1	110	2023-04-25
3	LENA	OPERATION	TEST_1	111	2023-04-19
4	VLADISLAV	OPERATION	HW_2	112	2023-04-17
5	LENA	OPERATION	HW_2	113	2023-04-23

Total rows: 5 of 5

Query complete 00:00:00.723

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











Код запроса:

```
create view "Uchet".project_view  
  
as select project.name_of_project, count(task.id_task) from  
  
"Uchet".project, "Uchet".task  
  
where project.id_project = task.id_project and task.task_status not in ('Завершен') and  
  
current_date = task.deadline_of_task  
  
group by project.id_project  
  
having count(task.id_task) >= 3 order by count(task.id_task) DESC
```

Результат запроса:

```
1 select * from "Uchet".project_view
```

Data Output Messages Notifications

							
name_of_project character (50) 		count bigint 					

Total rows: 0 of 0 Query complete 00:00:00.201

Задание 3. Запросы DELETE UPDATE INSERT

Запрос DELETE:

Удалить сотрудника, который не имеет задания

Код запроса:

```
delete from "Uchet".employee_in_department where id_employee_in_department not in (  
select task.id_employee_in_department from "Uchet".employee_in_department, "Uchet".task where  
employee_in_department.id_employee_in_department <> task.id_employee_in_department)
```

—

select * from "Uchet".employee_in_department

Результат до:

	id_employee_in_department [PK] integer	hours_of_work character (20)	salary integer	id_department integer	id_employee integer	id_positions integer
1+	[default]	[null]	[null]	[default]	[default]	[default]
2	11	30	40000	1	123	1
3	12	35	30000	1	234	2
4	13	36	40000	2	345	1
5	14	37	30000	2	456	2
6	15	34	40000	3	567	1
7	16	33	30000	3	678	2

Результат после:

	id_employee_in_department [PK] integer	hours_of_work character (20)	salary integer	id_department integer	id_employee integer	id_positions integer
1	14	37	30000	2	456	2
2	13	36	40000	2	345	1

Запрос UPDATE:

Повысить зарплату всем, у кого она меньше 40000, на 5000

Код запроса:

```
update "Uchet".positions set salary_of_position = salary_of_position + 5000 WHERE salary_of_position  
IN (SELECT salary_of_position FROM "Uchet".positions WHERE salary_of_position < 40000)
```

select * from "Uchet".positions

Результат до:

	id_position [PK] integer	name_of_position character (50)	salary_of_position integer
1	1	HEAD OF DEPARTMENT	40000
2	2	WORKER	30000

Результат после:

	id_position [PK] integer	name_of_position character (50)	salary_of_position integer
1	1	HEAD OF DEPARTMENT	40000
2	2	WORKER	35000

Запрос INSERT:

Добавление новых сотрудников

Код запроса:

```
insert into "Uchet".employee (id_employee, name_of_employee, contacts_of_employee)
```

```
VALUES (777, 'VITYA', (87777777777))
```

```
Select * from "Uchet".employee
```

Результат до:

	id_employee [PK] integer	name_of_employee character (50)	contacts_of_employee character (11)
1	567	ALFRED	89567567567
2	456	LENA	89456456456
3	345	VLADISLAV	89345345345
4	234	FEDOR	89888888888
5	123	DIMA	89825832761
6	678	GEORGE	89678678678
7	890	GOSHA	89889898889

Результат после:

	id_employee [PK] integer	name_of_employee character (50)	contacts_of_employee character (11)
1	567	ALFRED	89567567567
2	456	LENA	89456456456
3	345	VLADISLAV	89345345345
4	234	FEDOR	89888888888
5	123	DIMA	89825832761
6	678	GEORGE	89678678678
7	890	GOSHA	89889898889
8	777	VITYA	87777777777

Задание 4 История запросов

Today - 01.06.2023
<pre>SELECT * FROM "Uchet".employee ORDER BY id_employee ASC</pre> 03:03:06
<pre>select * from "Uchet".employee</pre> 02:52:13
<pre>insert into "Uchet".employee (id_employee, name_of_employee, cont...</pre> 02:51:34
<pre>select * from "Uchet".employee</pre> 02:51:03
<pre>select * from "Uchet".positions</pre> 02:46:41
<pre>update "Uchet".positions set salary_of_position = salary_of_posit...</pre> 02:45:50
<pre>select * from "Uchet".positions</pre> 02:43:45
<pre>select * from "Uchet".position</pre> 02:43:40
<pre>select * from "Uchet".employee_in_department</pre> 02:26:35
<pre>delete from "Uchet".employee_in_department where id_employee_in_d...</pre> 02:25:50
<pre>DELETE FROM "Uchet".stage WHERE id_stage IN (1134);</pre> 02:24:59
<pre>SELECT * FROM "Uchet".stage ORDER BY id_stage ASC</pre> 02:24:59
<pre>delete from "Uchet".employee_in_department where id_employee_in_d...</pre> 02:24:24
<pre>delete from "Uchet".employee_in_departmnet where id_employee_in_d...</pre> 02:24:09
<pre>delete from "Uchet".stage where stage_status not in (select stag...</pre>
Query Query History
Show queries generated internally by pgAdmin? <input checked="" type="checkbox"/> <div>Remove Remove All</div>
Today - 01.06.2023
<pre>select * from "Uchet".positions</pre> 02:43:45
<pre>select * from "Uchet".position</pre> 02:43:40
<pre>select * from "Uchet".employee_in_department</pre> 02:26:35
<pre>delete from "Uchet".employee_in_department where id_employee_in_d...</pre> 02:25:50
<pre>DELETE FROM "Uchet".stage WHERE id_stage IN (1134);</pre> 02:24:59
<pre>SELECT * FROM "Uchet".stage ORDER BY id_stage ASC</pre> 02:24:59
<pre>delete from "Uchet".employee_in_department where id_employee_in_d...</pre> 02:24:24
<pre>delete from "Uchet".employee_in_departmnet where id_employee_in_d...</pre> 02:24:09
<pre>delete from "Uchet".stage where stage_status not in (select stag...</pre> 02:19:41
<pre>delete from "Uchet".stage where stage_status in (select stage.id...</pre> 02:19:32
<pre>delete from "Uchet".stage where stage_status in (select stage.id...</pre> 02:19:25
<pre>SELECT * FROM "Uchet".stage ORDER BY id_stage ASC</pre> 02:16:16
<pre>delete from "Uchet".employee where id_employee not in (select ta...</pre> 02:14:04
<pre>select * from "Uchet".employee</pre> 02:13:31


Задание 6 EXPLAIN

Без индексов 2-й запрос

Код запроса:

```
explain select id_project, name_of_project, date_of_start from "Uchet".project  
where current_date - date_of_start > 30 order by id_project
```

Результат запроса:


	QUERY PLAN	
	text	
1	Sort (cost=18.57..18.82 rows=100 width=212)	
2	Sort Key: id_project	
3	-> Seq Scan on project (cost=0.00..15.25 rows=100 width=212)	
4	Filter: ((CURRENT_DATE - date_of_start) > 30)	
Total rows: 4 of 4 Query complete 00:00:00.066		

Без индексов 3-й запрос

Код запроса:

```
explain SELECT * FROM "Uchet".employee_in_department  
WHERE salary > (SELECT AVG(salary) FROM "Uchet".employee_in_department);
```

Результат запроса:

	QUERY PLAN	
	text	
1	Seq Scan on employee_in_department (cost=17.64..36.79 rows=203 width=104)	
2	Filter: ((salary)::numeric > \$0)	
3	InitPlan 1 (returns \$0)	
4	-> Aggregate (cost=17.63..17.64 rows=1 width=32)	
5	-> Seq Scan on employee_in_department employee_in_department_1 (cost=0.00..16.10 rows=610 width=4)	
Total rows: 5 of 5 Query complete 00:00:00.056		

Задание 7 Индексы

Создание простого индекса для 2-го запроса

Код запроса:

```
create index my_proj ON "Uchet".project(id_project)
```

Результат запроса:

```
1 create index my_proj ON "Uchet".project(id_project)
```

The screenshot shows a database management tool interface. At the top, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is currently selected and highlighted with a blue underline. Below the tabs, the text 'CREATE INDEX' is displayed. Further down, a status message reads 'Query returned successfully in 141 msec.'.

Создание составного индекса для 3-го запроса

Код запроса:

```
CREATE INDEX salary_department_index ON "Uchet".employee_in_department (salary, id_department);
```

Результат запроса:

```
1 CREATE INDEX salary_department_index ON "Uchet".employee_in_department (salary, id_department);
```

Data Output Messages Notifications


CREATE INDEX

Query returned successfully in 43 msec.

Total rows: 4 of 4 Query complete 00:00:00.043

Запросы с индексами

План 2-го запроса:

	QUERY PLAN	
	text	
1	Sort (cost=1.07..1.07 rows=1 width=212)	
2	Sort Key: id_project	
3	-> Seq Scan on project (cost=0.00..1.06 rows=1 width=212)	
4	Filter: ((id_project > 0) AND ((CURRENT_DATE - date_of_start) > 30))	
<div>Total rows: 4 of 4 Query complete 00:00:00.052</div>		

План 3-го запроса:

```

1 explain SELECT *
2 FROM "Uchet".employee_in_department
3 WHERE salary > (SELECT AVG(salary) FROM "Uchet".employee_in_department)
4 ORDER BY id_department;

```

Data Output Messages Notifications



	QUERY PLAN	
	text	
1	Sort (cost=2.08..2.08 rows=1 width=104)	
2	Sort Key: employee_in_department.id_department	
3	InitPlan 1 (returns \$0)	
4	-> Aggregate (cost=1.03..1.04 rows=1 width=32)	
5	-> Seq Scan on employee_in_department employee_in_department_1 (cost=0.00..1.02 rows=2 width=4)	
6	-> Seq Scan on employee_in_department (cost=0.00..1.03 rows=1 width=104)	
7	Filter: ((salary)::numeric > \$0)	

Total rows: 7 of 7 Query complete 00:00:00.064

Удаление индексов:

Удаление индекса для 2-го запроса:

```
Query Query History
1 drop index "Uchet".my_proj

Data Output Messages Notifications
DROP INDEX
Query returned successfully in 92 msec.
```

Удаление индекса для 3-го запроса:

```
1 drop index "Uchet".salary_department_index

Data Output Messages Notifications
DROP INDEX
Query returned successfully in 50 msec
```

Вывод:

SQL это очень удобная возможность быстро видоизменять 24 наши данные в базе данных. Мы можем изменять, добавлять или удалять данные, делать выборки по нужным нам критериям и считать нужные данные по необходимым характеристикам. Сравним время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись немного медленнее из-за небольшого количества данных в таблице.