

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет инфокоммуникационных технологий

Дисциплина:

«Проектирование и реализация баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.2

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

Выполнила:

Арнаутова Елизавета
группа К32422

Проверила:

Говорова Марина Михайловна

Санкт-Петербург
2023

ЛАБОРАТОРНАЯ РАБОТА №2

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL

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.

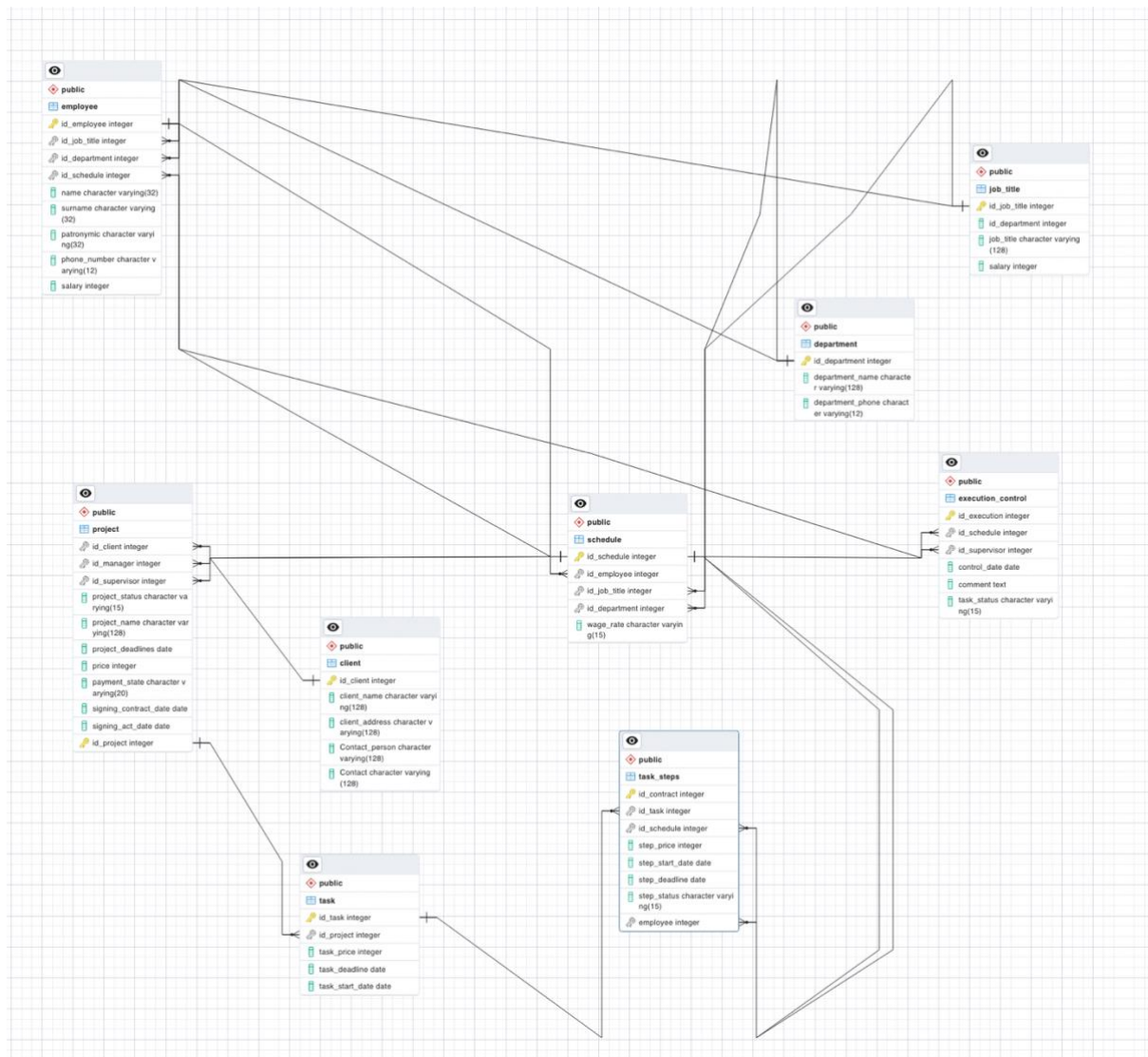


Рис. 1 – Схема базы данных

SELECT

Задание 1: создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).

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

```
select p.id_project, p.project_name, id_client, t.id_task, ts.employee, schedule.id_department
from project as p
left outer join task as t on p.id_project = t.id_project
left outer join task_steps as ts on ts.id_task = t.id_task
left outer join schedule on schedule.id_schedule = ts.employee
order by p.id_project;
```

	id_project integer	project_name character varying (128)	id_client integer	id_task integer	employee integer	id_department integer
1	1	коренной курс	45	25	11	271
2	1	коренной курс	45	410	178	463
3	1	коренной курс	45	410	3	285
4	1	коренной курс	45	25	235	418
5	1	коренной курс	45	410	302	484
6	1	коренной курс	45	410	31	444
7	1	коренной курс	45	410	416	225
8	1	коренной курс	45	410	329	258
9	1	коренной курс	45	410	27	211
10	1	коренной курс	45	25	458	167
11	2	атомный земля	132	74	148	36
12	2	атомный земля	132	376	34	153
13	2	атомный земля	132	74	1	366
14	2	атомный земля	132	149	186	125
15	2	атомный земля	132	376	366	191
16	2	атомный земля	132	74	50	252
17	2	атомный земля	132	376	272	374
18	3	банальный итог	26	206	332	103
19	4	экспертный характеристика	223	94	459	334
20	4	экспертный характеристика	223	414	358	49
21	4	экспертный характеристика	223	94	58	198
22	4	экспертный характеристика	223	414	83	249
23	4	экспертный характеристика	223	414	53	349
24	4	экспертный характеристика	223	94	112	87
25	4	экспертный характеристика	223	94	349	67
26	4	экспертный характеристика	223	95	[null]	[null]
27	4	экспертный	223	95	[null]	[null]
28	4	экспертный	223	95	[null]	[null]

✓ Successfully run. Total query runtime: 149 msec. 6903 rows affected. ✕

Рис. 2 – SELECT 1

SELECT

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

```
select * from project
where signing_contract_date > (current_date - interval '1 month')
and project_status != 'не начат';
```

Data Output Messages Notifications							
	id_client integer	id_manager integer	id_supervisor integer	project_status character varying (15)	project_name character varying (128)	project_deadlines date	
1	45	115	315	приостановлен	коренной курс	2010-06-22	
2	132	230	345	приостановлен	атомный земля	2059-08-09	
3	384	53	429	приостановлен	деловой документ	2032-10-18	
4	325	141	435	выполнен	действительный самолет	2031-04-24	
5	293	71	400	выполнен	загадочный президент	2073-11-04	
6	480	105	329	выполнен	прогрессивный соглашение	2040-02-17	
7	148	96	436	в процессе	универсальный читатель	2062-09-22	
8	347	239	476	в процессе	чрезвычайный рука	2061-12-18	
9	224	195	487	в процессе	вчерашний изображение	2044-10-17	
10	410	138	481	выполнен	органический сюжет	2079-02-28	
11	70	141	366	в процессе	великолепный данные	2083-10-06	
12	432	45	473	в процессе	эстетический граница	2000-01-01	
13	260	249	489	выполнен	звездный мальчишка	2048-12-20	
14	150	15	257	выполнен	ужасный сигнал	2067-02-18	
15	281	209	273	в процессе	последующий достижение	2085-10-17	
16	356	113	362	приостановлен	сухой ребенок	2067-11-22	
17	120	100	500	выполнен	криминальный крик	2046-10-21	
18	409	224	419	приостановлен	позитивный письмо	2034-07-01	
19	471	141	380	выполнен	советский бизнес	2051-09-08	
20	84	64	406	выполнен	голодный источник	2053-06-25	
21	25	106	311	выполнен	умственный лошадь	2057-03-11	
22	104	143	257	в процессе	милицейский бог	2085-04-19	

ery complete 00:00:00.079

✓ Successfully run. Total query runtime: 79 msec. 915 rows affected. ✕

Рис. 3 – SELECT 2

SELECT

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

```
select id_employee, employee.id_department, salary, dep_average
from employee
left join
(select id_department, avg(salary) as dep_average
from employee
group by id_department) as employee_2
on employee.id_department = employee_2.id_department
where salary > dep_average
order by employee.id_department
```

Data Output					Messages	Notifications
	id_employee [PK] integer	id_department integer	salary integer	dep_average numeric		
1	374	1	256709469	201268965.00000000		
2	378	2	750702438	519946581.00000000		
3	269	5	133891945	87864005.333333333333		
4	24	7	333789190	293119469.50000000		
5	466	10	532952887	458792579.00000000		
6	175	10	698046306	458792579.00000000		
7	309	17	604838673	366932775.00000000		
8	28	21	650980327	338742662.50000000		
9	310	33	621598244	457244250.33333333		
10	399	35	593886101	506271245.75000000		
11	94	35	704897518	506271245.75000000		
12	6	37	279767916	236963543.00000000		
13	201	40	381901540	205130570.50000000		
14	332	43	717683541	493901136.50000000		
15	321	43	795839370	493901136.50000000		
16	433	46	618348217	604427113.00000000		
17	168	56	644082078	336533424.50000000		
18	115	57	738280906	544230985.50000000		
19	65	60	695458466	395046422.00000000		
20	110	65	434131914	383970814.00000000		
21	55	66	434893427	366168255.40000000		
22	2	66	650022427	366168255.40000000		
23	320	69	504882197	321893065.00000000		
24	53	71	734755654	551083075.00000000		
25	480	77	617678610	382305954.50000000		
26	92	79	690661179	554153591.00000000		
27		
28		

✓ Successfully run. Total query runtime: 93 msec. 151 rows affected. ✕

Ln 14, Col 1

Рис. 4 – SELECT 3

SELECT

4. Найти отдел, работающий над максимальным количеством проектов.

```
select s.id_department, d.department_name, count(s.id_department) as count_d
from task_steps, schedule as s, department as d
where employee = s.id_schedule and d.id_department = s.id_department
group by s.id_department, d.department_name
order by count_d desc
limit 5
```

Data Output Messages Notifications			
	id_department integer	department_name character varying (128)	count_d bigint
1	287	IT-отдел	30
2	124	Отдел продаж	24
3	470	Отдел продаж	24
4	317	Отдел технической поддержки	24
5	125	Администрация	24

Рис. 5 – SELECT 4

SELECT

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

```
select e.id_employee, name, surname, patronymic, phone_number,
t.id_project, t.id_task, t.task_deadline
from employee as e
left join task_steps as ts on e.id_employee = ts.employee
left join task as t on ts.id_task = t.id_task
order by e.id_employee
```

Data Output Messages Notifications							
	id_employee integer	name character varying (32)	surname character varying (32)	patronymic character varying (32)	phone_number character varying (12)	id_project integer	i
1	1	Макарова	Елизавета	Германовна	83344758008	206	
2	1	Макарова	Елизавета	Германовна	83344758008	2870	
3	1	Макарова	Елизавета	Германовна	83344758008	2534	
4	1	Макарова	Елизавета	Германовна	83344758008	477	
5	1	Макарова	Елизавета	Германовна	83344758008	2	
6	2	Малышев	Денис	Денисович	954603998538	88	
7	2	Малышев	Денис	Денисович	954603998538	2511	
8	2	Малышев	Денис	Денисович	954603998538	285	
9	2	Малышев	Денис	Денисович	954603998538	2419	
10	2	Малышев	Денис	Денисович	954603998538	2603	
11	2	Малышев	Денис	Денисович	954603998538	35	
12	3	Еремин	Дмитрий	Артёмович	109201314829	2578	
13	3	Еремин	Дмитрий	Артёмович	109201314829	205	
14	3	Еремин	Дмитрий	Артёмович	109201314829	2677	
15	3	Еремин	Дмитрий	Артёмович	109201314829	1	
16	3	Еремин	Дмитрий	Артёмович	109201314829	184	
17	3	Еремин	Дмитрий	Артёмович	109201314829	2753	
18	4	Губанова	Виктория	Михайловна	382565178636	2605	
19	4	Губанова	Виктория	Михайловна	382565178636	2911	
20	4	Губанова	Виктория	Михайловна	382565178636	450	
21	4	Губанова	Виктория	Михайловна	382565178636	148	
22	4	Губанова	Виктория	Михайловна	382565178636	2416	
23	4	Губанова	Виктория	Михайловна	382565178636	465	
24	5	Самойлова	Елизавета	Артёмьевна	72871452905	2781	
25	5	Самойлова	Елизавета	Артёмьевна	72871452905	2769	
26	5	Самойлова	Елизавета	Артёмьевна	72871452905	263	
27	5	Самойлова	Елизавета	Артёмьевна	72871452905	263	

✓ Successfully run. Total query runtime: 138 msec. 2993 rows affected.
✗

Query complete 00:00:00.138

Ln 10, Col 1

Рис. 6 – SELECT 5

SELECT

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.

```
select e.id_employee, t.id_project, t.id_task, t.task_deadline,  
step_status, (current_date - task_deadline) as delay  
from employee as e  
left join task_steps as ts on e.id_employee = ts.employee  
left join task as t on ts.id_task = t.id_task  
where step_status in ('в процессе', 'не начат')  
and current_date > task_deadline  
order by e.id_employee
```

Data Output Messages Notifications						
	id_employee integer	id_project integer	id_task integer	task_deadline date	step_status character varying (15)	delay integer
1	2	2511	1778	2023-03-06	не начат	50
2	3	2677	1344	2023-04-07	не начат	18
3	4	2911	1105	2023-02-06	не начат	78
4	6	379	482	2009-12-19	не начат	4875
5	8	31	113	2013-07-20	в процессе	3566
6	8	340	479	2011-10-21	не начат	4204
7	9	500	7	2012-06-05	в процессе	3976
8	10	251	388	2015-11-10	в процессе	2723
9	11	134	176	2005-09-02	в процессе	6444
10	12	188	104	2016-11-05	в процессе	2362
11	14	2950	1887	2023-01-13	не начат	102
12	16	2894	1744	2023-02-18	не начат	66
13	16	2658	962	2023-01-21	не начат	94
14	18	2621	1343	2023-04-06	в процессе	19
15	18	2503	1535	2023-03-07	в процессе	49
16	19	2611	1267	2023-02-20	не начат	64
17	20	232	229	2013-08-05	не начат	3550
18	21	2799	1364	2023-03-06	не начат	50
19	22	60	456	2013-08-16	не начат	3539
20	23	2908	1626	2023-02-24	в процессе	60
21	23	2777	1063	2023-02-08	не начат	76
22	24	55	398	2022-09-01	не начат	236
23	25	2651	1354	2023-01-16	в процессе	99
24	29	2418	984	2023-04-15	в процессе	10
25	30	304	111	2009-01-07	в процессе	5221
26	31	2415	1703	2023-02-27	не начат	57
27	32	375	888	2006-02-11	не начат	6458
28	32	29	29	2023-02-11	не начат	6458

✓ Successfully run. Total query runtime: 74 msec. 407 rows affected. ✕

74Ln 14, Col 1

Рис. 7 – SELECT 6

SELECT

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

```
select p.id_project, p.project_name, count(ts.employee) as employee_number
from project as p
left outer join task as t on p.id_project = t.id_project
left outer join task_steps as ts on ts.id_task = t.id_task
left outer join schedule on schedule.id_schedule = ts.employee
group by p.id_project
having count(ts.employee) > 3
order by employee_number desc
```

Data Output Messages Notifications			
	id_project [PK] integer	project_name character varying (128)	employee_number bigint
1	105	предметный правда	18
2	65	действующий предприятие	16
3	389	погибший сравнение	15
4	280	заклученный выступление	14
5	275	характерный будущее	13
6	192	подводной правда	13
7	481	подводной орган	13
8	134	худой сын	12
9	2656	этнический лев	12
10	2790	рыбный отец	12
11	340	вынужденный достоинство	12
12	361	краткий дача	11
13	38	весенний рост	11
14	158	женский зритель	11
15	2440	неведомый стиль	11
16	402	знаменитый аппарат	11
17	346	должен май	11
18	396	специфический ящик	11
19	2511	небесный должность	11
20	293	сельскохозяйственный старик	11
21	289	радикальный затрата	11
22	213	служебный сбор	11
23	2908	оперативный житель	11
24	115	смешной основа	10
25	96	плотный улица	10
26	2799	парижский бой	10
27	8181	*	10
28	✓ Successfully run. Total query runtime: 102 msec. 357 rows affected. ✕		

Ln 13, Col 1

Рис. 8 – SELECT 7

VIEW

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

```
create or replace view project_info as
select p.id_project, p.project_name, t.id_task, t.task_deadline,
s.id_department, ts.employee, e.name, e.surname, e.patronymic, e.phone_number
from project as p
left outer join task as t on p.id_project = t.id_project
left outer join task_steps as ts on ts.id_task = t.id_task
left outer join schedule as s on s.id_schedule = ts.employee
left outer join employee as e on s.id_schedule = e.id_employee
order by t.id_task, ts.employee;
```

Data Output

Messages

Notifications

	id_project integer	project_name character varying (128)	id_task integer	task_deadline date	id_department integer	employee integer	name character varying (32)	surname character varying (32)	patronymic character varying (32)
1	105	предметный правда	252	2079-07-02	29	25	Нечаева	Аиша	Артёмовна
2	105	предметный правда	252	2079-07-02	317	105	Кулагин	Ярослав	Арсентьевич
3	105	предметный правда	252	2079-07-02	22	162	Павловский	Лука	Андреевич
4	105	предметный правда	252	2079-07-02	168	214	Кулагин	Ярослав	Арсентьевич
5	105	предметный правда	252	2079-07-02	228	248	Иванов	Александр	Михайлович
6	105	предметный правда	252	2079-07-02	208	250	Смирнова	Елизавета	Алиевна
7	105	предметный правда	252	2079-07-02	253	439	Григорьева	Виктория	Сергеевна
8	105	предметный правда	252	2079-07-02	307	449	Петров	Семён	Максимович
9	105	предметный правда	306	2056-02-22	425	68	Ефимов	Николай	Кириллович
10	105	предметный правда	306	2056-02-22	287	109	Абрамова	Валерия	Матвеевна
11	105	предметный правда	306	2056-02-22	72	267	Волкова	Дарина	Дмитриевна
12	105	предметный правда	306	2056-02-22	190	473	Кошелева	Марьяна	Александровна
13	105	предметный правда	328	2029-08-08	285	63	Волкова	Дарья	Никитична
14	105	предметный правда	328	2029-08-08	314	99	Власова	Мирослава	Билаловна
15	105	предметный правда	328	2029-08-08	359	209	Семенов	Марк	Матвеевич
16	105	предметный правда	328	2029-08-08	140	300	Леонова	Арина	Кирилловна
17	105	предметный правда	328	2029-08-08	264	350	Никулина	Милана	Тимуровна
18	105	предметный правда	328	2029-08-08	225	403	Горелов	Роман	Сергеевич

✓ Successfully run. Total query runtime: 118 msec. 18 rows affected. ✕

complete 00:00:00.118

Ln 11, Col 1

Рис. 9 – VIEW 1

VIEW

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

```
create or replace view deadline_today as
select project.id_project, project_deadlines,
count(task.id_task) as unexecuted, id_supervisor
from project
left join task on task.id_project = project.id_project
left join task_steps on task.id_task = task_steps.id_task
where current_date = project_deadlines and step_status != 'выполнен'
group by project.id_project, task.id_task
having count(task.id_task) >= 3
order by project.id_project
```













Data Output Messages Notifications				
       				
	id_project integer 	project_deadlines date 	unexecuted bigint 	id_supervisor integer 
1	2936	2023-04-25	3	343
2	2944	2023-04-25	3	341

Рис. 10 – VIEW 2

INSERT

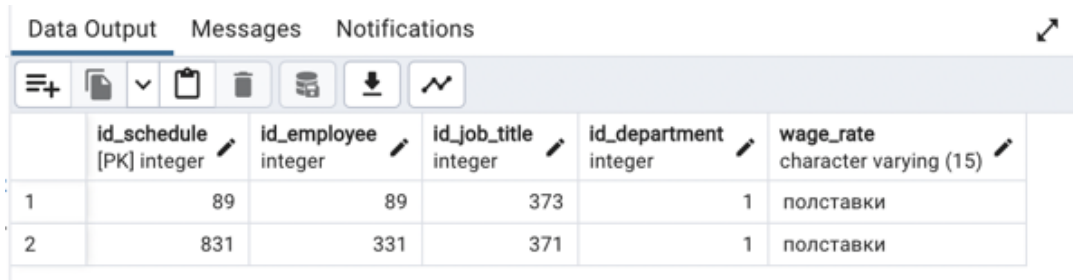
Задание 2: составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

Что делает? Собирает всех лаборантов, у которых не больше двух работ (чтоб не перерабатывали!), и отправляет в отдел инновационных разработок на полставки.

```
insert into schedule (id_employee, id_job_title, id_department, wage_rate)
select id_employee, schedule.id_job_title, 1 as new_department, 'полставки' as new_rate
from job_title
left join schedule on schedule.id_job_title = job_title.id_job_title
where job_title = 'Лаборант'
and wage_rate = 'полная ставка'
and id_employee in
(select id_employee from schedule
group by id_employee
having count(id_employee) <= 2)
```

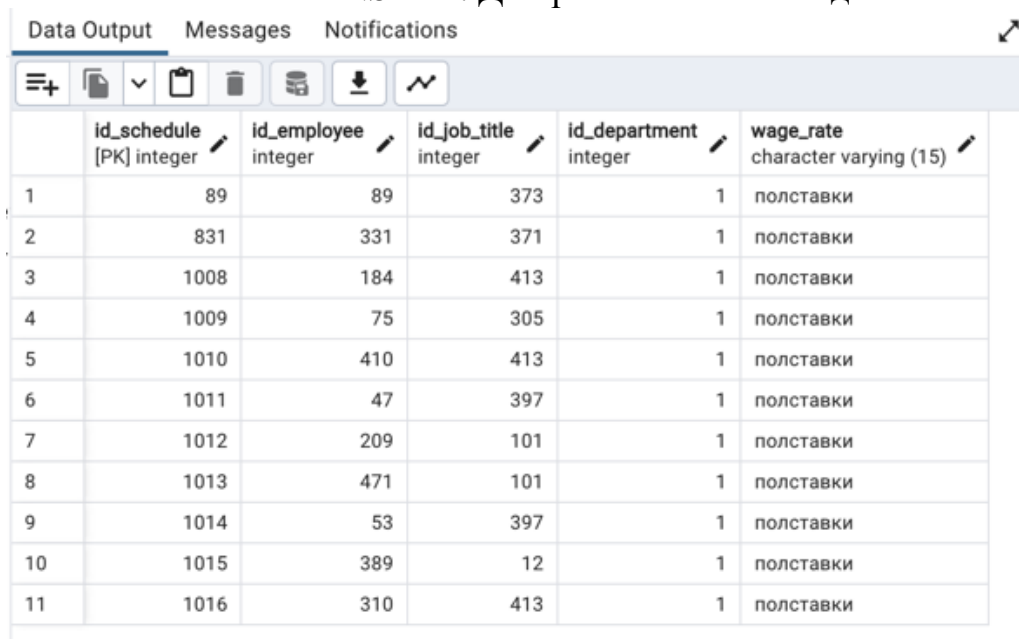
Результаты приводятся для запроса

```
select * from schedule where id_department = 1
```



	id_schedule [PK] integer	id_employee integer	id_job_title integer	id_department integer	wage_rate character varying (15)
1	89	89	373	1	полставки
2	831	331	371	1	полставки

Рис. 11 – INSERT. До применения команды



	id_schedule [PK] integer	id_employee integer	id_job_title integer	id_department integer	wage_rate character varying (15)
1	89	89	373	1	полставки
2	831	331	371	1	полставки
3	1008	184	413	1	полставки
4	1009	75	305	1	полставки
5	1010	410	413	1	полставки
6	1011	47	397	1	полставки
7	1012	209	101	1	полставки
8	1013	471	101	1	полставки
9	1014	53	397	1	полставки
10	1015	389	12	1	полставки
11	1016	310	413	1	полставки

Рис. 12 – INSERT. После применения команды

UPDATE

Что делает? Для (хотя бы частично) оплаченных, но приостановленных проектов, которые заказали несколько определенных организаций, статус заказа меняется с «приостановлен» на «в процессе»

```
update task_steps
set step_status = 'в процессе'
where id_contract in
(select id_contract from task_steps
left join task on task_steps.id_task = task.id_task
left join project on task.id_project = project.id_project
left join client on project.id_client = client.id_client
where step_status = 'приостановлен'
and client_name in ('рыбный читатель', 'полноценный банк', 'жалкий показатель', 'автономный сын', 'дипломатический лид')
and payment_state != 'не оплачен'
order by client.id_client
)
```

Результаты привожу для вот этого запроса:

```
select * from task_steps where step_status = 'приостановлен'
```

Видно, что после его выполнения количество «приостановленных» проектов уменьшилось

	id_contract [PK] integer	id_task integer	id_schedule integer	step_price integer	step_start_date date	step_deadline date	step_status character varying (15)	employee integer
1	1	74	38	152074	2001-02-16	2003-09-11	приостановлен	1
2	2	190	362	315762	2074-12-06	2077-08-21	приостановлен	2
3	3	191	281	172142	2038-01-18	2040-08-25	приостановлен	3
4	4	178	100	119622	2025-10-13	2027-08-14	приостановлен	4
5	5	365	100	475899	2029-05-06	2030-05-14	приостановлен	5
6	6	13	393	453782	2058-06-01	2061-10-09	приостановлен	6
7	7	435	230	407159	2024-12-08	2027-03-18	приостановлен	7
8	8	460	371	190665	2057-11-06	2060-10-26	приостановлен	8
9	9	53	474	350001	2074-12-01	2076-01-08	приостановлен	9
10	10	472	248	296109	2038-04-18	2041-11-19	приостановлен	10
11	11	25	146	77599	2009-12-01	2012-02-04	приостановлен	11
12	12	314	140	313332	2083-03-22	2085-11-04	приостановлен	12
13	13	15	306	285203	2030-07-17	2032-09-14	приостановлен	13
14	14	203	432	8637	2078-02-15	2079-08-02	приостановлен	14
15	15	214	446	221227	2039-01-02	2040-11-16	приостановлен	15
16	16	56	287	86267	2060-08-22	2061-04-08	приостановлен	16
17	17	84	305	159509	2041-10-12	2042-12-25	приостановлен	17
18	18	192	198	442245	2042-03-03	2043-07-03	приостановлен	18
19	19	42	34	153187	2029-03-27	2032-09-12	приостановлен	19
20	20	102	327	126753	2037-12-27	2039-01-16	приостановлен	20
21	21	176	130	52493	2090-08-24	2091-02-16	приостановлен	21
22	22	68	211	466963	2000-08-16	2003-01-25	приостановлен	22
23	23	360	257	75099	2019-09-08	2021-12-09	приостановлен	23
24	24	412	491	271690	2035-02-21	2037-04-14	приостановлен	24
25	25	37	156	418538	2008-10-14	2009-09-07	приостановлен	25
26	26	237	342	451056	2067-01-11	2069-07-12	приостановлен	26
27	27	101	463	412137	2000-10-08	2000-11-08	приостановлен	27
28	28	85	436	47760	2000-10-08	2000-11-08	приостановлен	28

✓ Successfully run. Total query runtime: 83 msec. 1109 rows affected.

query complete 00:00:00.083Ln 14, Col 61

UPDATE

Рис. 13 – UPDATE. До применения команды

	id_contract [PK] integer	id_task integer	id_schedule integer	step_price integer	step_start_date date	step_deadline date	step_status character varying (15)	employee integer
1	1	74	38	152074	2001-02-16	2003-09-11	приостановлен	1
2	2	190	362	315762	2074-12-06	2077-08-21	приостановлен	2
3	3	191	281	172142	2038-01-18	2040-08-25	приостановлен	3
4	4	178	100	119622	2025-10-13	2027-08-14	приостановлен	4
5	5	365	100	475899	2029-05-06	2030-05-14	приостановлен	5
6	6	13	393	453782	2058-06-01	2061-10-09	приостановлен	6
7	7	435	230	407159	2024-12-08	2027-03-18	приостановлен	7
8	8	460	371	190665	2057-11-06	2060-10-26	приостановлен	8
9	9	53	474	350001	2074-12-01	2076-01-08	приостановлен	9
10	10	472	248	296109	2038-04-18	2041-11-19	приостановлен	10
11	11	25	146	77599	2009-12-01	2012-02-04	приостановлен	11
12	12	314	140	313332	2083-03-22	2085-11-04	приостановлен	12
13	13	15	306	285203	2030-07-17	2032-09-14	приостановлен	13
14	14	203	432	8637	2078-02-15	2079-08-02	приостановлен	14
15	15	214	446	221227	2039-01-02	2040-11-16	приостановлен	15
16	16	56	287	86267	2060-08-22	2061-04-08	приостановлен	16
17	17	84	305	159509	2041-10-12	2042-12-25	приостановлен	17
18	18	192	198	442245	2042-03-03	2043-07-03	приостановлен	18
19	19	42	34	153187	2029-03-27	2032-09-12	приостановлен	19
20	20	102	327	126753	2037-12-27	2039-01-16	приостановлен	20
21	21	176	130	52493	2090-08-24	2091-02-16	приостановлен	21
22	22	68	211	466963	2000-08-16	2003-01-25	приостановлен	22
23	23	360	257	75099	2019-09-08	2021-12-09	приостановлен	23
24	24	412	491	271690	2035-02-21	2037-04-14	приостановлен	24
25	25	37	156	418538	2008-10-14	2009-09-07	приостановлен	25
26	26	237	342	451056	2067-01-11	2069-07-12	приостановлен	26
27	27	101	463	412137	2000-10-02	2000-11-02	приостановлен	27
28	28	85	436	47760	2000-10-02	2000-11-02	приостановлен	28

query complete 00:00:00.057

Ln 14, Col 44

✓ Successfully run. Total query runtime: 57 msec. 1123 rows affected.

Рис. 14 – UPDATE. После применения команды

DELETE

Что делает? Все задания, которыми занимается IT-отдел, дедлайн которых зачем-то назначен на через 50 лет и статус «приостановлен» подлежат уничтожению

```
delete from task_steps where id_contract in
(select id_contract from task_steps
left join employee on task_steps.employee = employee.id_employee
left join department on department.id_department = employee.id_department
where
step_deadline > current_date + interval '50 year'
and step_status = 'приостановлен'
and department_name = 'IT-отдел')
```

Тут — аналогично пункту update


```
select * from task_steps
```

Просто общая выборка по столбцу заданий.

DELETE

	id_contract [PK] integer	id_task integer	id_schedule integer	step_price integer	step_start_date date	step_deadline date	step_status character varying (15)	employee integer
1	1	74	38	152074	2001-02-16	2003-09-11	приостановлен	1
2	2	190	362	315762	2074-12-06	2077-08-21	приостановлен	2
3	3	191	281	172142	2038-01-18	2040-08-25	приостановлен	3
4	4	178	100	119622	2025-10-13	2027-08-14	приостановлен	4
5	5	365	100	475899	2029-05-06	2030-05-14	приостановлен	5
6	6	13	393	453782	2058-06-01	2061-10-09	приостановлен	6
7	7	435	230	407159	2024-12-08	2027-03-18	приостановлен	7
8	8	460	371	190665	2057-11-06	2060-10-26	приостановлен	8
9	9	53	474	350001	2074-12-01	2076-01-08	приостановлен	9
10	10	472	248	296109	2038-04-18	2041-11-19	приостановлен	10
11	11	25	146	77599	2009-12-01	2012-02-04	приостановлен	11
12	12	314	140	313332	2083-03-22	2085-11-04	приостановлен	12
13	13	15	306	285203	2030-07-17	2032-09-14	приостановлен	13
14	14	203	432	8637	2078-02-15	2079-08-02	приостановлен	14
15	15	214	446	221227	2039-01-02	2040-11-16	приостановлен	15
16	16	56	287	86267	2060-08-22	2061-04-08	приостановлен	16
17	17	84	305	159509	2041-10-12	2042-12-25	приостановлен	17
18	18	192	198	442245	2042-03-03	2043-07-03	приостановлен	18
19	19	42	34	153187	2029-03-27	2032-09-12	приостановлен	19
20	20	102	327	126753	2037-12-27	2039-01-16	приостановлен	20
21	21	176	130	52493	2090-08-24	2091-02-16	приостановлен	21
22	22	68	211	466963	2000-08-16	2003-01-25	приостановлен	22
23	23	360	257	75099	2019-09-08	2021-12-09	приостановлен	23
24	24	412	491	271690	2035-02-21	2037-04-14	приостановлен	24
25	25	37	156	418538	2008-10-14	2009-09-07	приостановлен	25
26	26	237	342	451056	2067-01-11	2069-07-12	приостановлен	26
27	27	101	463	412137	2022-12-22	2023-11-22	приостановлен	27

query complete 00:00:00.080

✓ Successfully run. Total query runtime: 80 msec. 3000 rows affected. ✕

Рис. 15 – DELETE. До применения команды

DELETE

	id_contract [PK] integer	id_task integer	id_schedule integer	step_price integer	step_start_date date	step_deadline date	step_status character varying (15)	employee integer
1	1	74	38	152074	2001-02-16	2003-09-11	приостановлен	1
2	2	190	362	315762	2074-12-06	2077-08-21	приостановлен	2
3	3	191	281	172142	2038-01-18	2040-08-25	приостановлен	3
4	4	178	100	119622	2025-10-13	2027-08-14	приостановлен	4
5	5	365	100	475899	2029-05-06	2030-05-14	приостановлен	5
6	6	13	393	453782	2058-06-01	2061-10-09	приостановлен	6
7	7	435	230	407159	2024-12-08	2027-03-18	приостановлен	7
8	8	460	371	190665	2057-11-06	2060-10-26	приостановлен	8
9	9	53	474	350001	2074-12-01	2076-01-08	приостановлен	9
10	10	472	248	296109	2038-04-18	2041-11-19	приостановлен	10
11	11	25	146	77599	2009-12-01	2012-02-04	приостановлен	11
12	12	314	140	313332	2083-03-22	2085-11-04	приостановлен	12
13	13	15	306	285203	2030-07-17	2032-09-14	приостановлен	13
14	14	203	432	8637	2078-02-15	2079-08-02	приостановлен	14
15	15	214	446	221227	2039-01-02	2040-11-16	приостановлен	15
16	16	56	287	86267	2060-08-22	2061-04-08	приостановлен	16
17	17	84	305	159509	2041-10-12	2042-12-25	приостановлен	17
18	18	192	198	442245	2042-03-03	2043-07-03	приостановлен	18
19	19	42	34	153187	2029-03-27	2032-09-12	приостановлен	19
20	20	102	327	126753	2037-12-27	2039-01-16	приостановлен	20
21	21	176	130	52493	2090-08-24	2091-02-16	приостановлен	21
22	22	68	211	466963	2000-08-16	2003-01-25	приостановлен	22
23	23	360	257	75099	2019-09-08	2021-12-09	приостановлен	23
24	24	412	491	271690	2035-02-21	2037-04-14	приостановлен	24
25	25	37	156	418538	2008-10-14	2009-09-07	приостановлен	25
26	26	237	342	451056	2067-01-11	2069-07-12	приостановлен	26
27	27	101	463	412137	2000-08-16	2003-01-25	приостановлен	27
28	28	85	436	47760	2000-08-16	2003-01-25	приостановлен	28

Query complete 00:00:00.083

Ln 1, Col 25

Successfully run. Total query runtime: 83 msec. 2993 rows affected.

Рис. 16 – DELETE. После применения команды

INDEX

Задание 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.

Для выполнения — взяла шестой и седьмой запросы из группы SELECT.

The screenshot displays a database query editor with a query window on the left and a query plan window on the right. The query in the left window is as follows:

```
1 create index manager_index on employee(id_employee)
2
3 explain
4 select e.id_employee, t.id_project, t.id_task, t.task_name
5 from employee as e
6 left join task_steps as ts on e.id_employee = ts.employee_id
7 left join task as t on ts.id_task = t.id_task
8 where step_status in ('в процессе', 'не начал')
9 order by e.id_employee
```

The query plan window on the right shows the execution plan for the query. The plan starts with a sequential scan on the employee table (cost=202.36..203.34 rows=394 width=41). This is followed by a sort operation (cost=0.00..0.00 rows=394 width=41). The plan then shows a hash join between employee and task_steps (cost=108.58..185.37 rows=394 width=41). This is followed by a hash join between task_steps and task (cost=89.33..163.11 rows=394 width=37). The plan then shows a hash join between task and task_name (cost=75.50..75.50 rows=1106 width=12). This is followed by a sequential scan on task_steps (cost=0.00..70.51 rows=1246 width=29). The plan then shows a filter operation (cost=0.00..0.00 rows=1106 width=12). This is followed by a sequential scan on task (cost=0.00..75.50 rows=1106 width=12). The plan then shows a filter operation (cost=13.00..13.00 rows=500 width=4). This is followed by a sequential scan on employee (cost=0.00..13.00 rows=500 width=4).

The status bar at the bottom of the editor indicates: "Total rows: 13 of 13 | Query complete 00:00:00.047 | Ln 11, Col 1". A green message box at the bottom right says: "Successfully run. Total query runtime: 47 msec. 13 rows affected. X".

Рис. 17 – INDEX. Шестой запрос. План без индексации

Query Query History

```

1 explain
2 select e.id_employee, t.id_project, t.id_task, t.task_status
3 from employee as e
4 left join task_steps as ts on e.id_employee = ts.employee
5 left join task as t on ts.id_task = t.id_task
6 where step_status in ('в процессе', 'не начал') and
7 order by e.id_employee
8
9

```

Data Output Messages Notifications

QUERY PLAN
text

1	Sort (cost=202.36..203.34 rows=394 width=41)
2	Sort Key: e.id_employee
3	-> Hash Join (cost=108.58..185.37 rows=394 width=41)
4	Hash Cond: (ts.employee = e.id_employee)
5	-> Hash Join (cost=89.33..163.11 rows=394 width=37)
6	Hash Cond: (ts.id_task = t.id_task)
7	-> Seq Scan on task_steps ts (cost=0.00..70.51 rows=1246 width=29)
8	Filter: ((step_status)::text = ANY ({"в процессе","не начал"}::text[]))
9	-> Hash (cost=75.50..75.50 rows=1106 width=12)
10	-> Seq Scan on task t (cost=0.00..75.50 rows=1106 width=12)
11	Filter: (CURRENT_DATE > task_deadline)
12	-> Hash (cost=13.00..13.00 rows=500 width=4)
13	-> Seq Scan on employee e (cost=0.00..13.00 rows=500 width=4)

✓ Successfully run. Total query runtime: 64 msec. 13 rows affected. ✕

Total rows: 13 of 13 Query complete 00:00:00.064 Ln 9, Col 1

Рис. 18 – INDEX. Шестой запрос. План с простым индексом

Query Query History

```

1 create index manager_index on employee(id_employee)
2 create index manager_project_index on task(id_project)
3 create index manager_task_index on task(id_task);
4
5 explain
6 select e.id_employee, t.id_project, t.id_task, t.task_status
7 from employee as e
8 left join task_steps as ts on e.id_employee = ts.employee
9 left join task as t on ts.id_task = t.id_task
10 where step_status in ('в процессе', 'не начал') and
11 order by e.id_employee
12
13

```

Data Output Messages Notifications

QUERY PLAN
text

1	Sort (cost=202.36..203.34 rows=394 width=41)
2	Sort Key: e.id_employee
3	-> Hash Join (cost=108.58..185.37 rows=394 width=41)
4	Hash Cond: (ts.employee = e.id_employee)
5	-> Hash Join (cost=89.33..163.11 rows=394 width=37)
6	Hash Cond: (ts.id_task = t.id_task)
7	-> Seq Scan on task_steps ts (cost=0.00..70.51 rows=1246 width=29)
8	Filter: ((step_status)::text = ANY ({"в процессе","не начал"}::text[]))
9	-> Hash (cost=75.50..75.50 rows=1106 width=12)
10	-> Seq Scan on task t (cost=0.00..75.50 rows=1106 width=12)
11	Filter: (CURRENT_DATE > task_deadline)
12	-> Hash (cost=13.00..13.00 rows=500 width=4)
13	-> Seq Scan on employee e (cost=0.00..13.00 rows=500 width=4)

✓ Successfully run. Total query runtime: 98 msec. 13 rows affected. ✕

Total rows: 13 of 13 Query complete 00:00:00.098 Ln 13, Col 1

Рис. 19 – INDEX. Шестой запрос. План с составным индексом

Query Query History Data Output Messages Notifications

```

1 explain
2 select p.id_project, p.project_name, count(ts.employee)
3 from project as p
4 left outer join task as t on p.id_project = t.id_project
5 left outer join task_steps as ts on ts.id_task = t.id_task
6 left outer join schedule on schedule.id_schedule = ts.id_schedule
7 group by p.id_project
8 having count(ts.employee) > 3
9 order by employee_number desc
10

```

QUERY PLAN

1	Sort (cost=380.66..382.82 rows=867 width=45)
2	Sort Key: (count(ts.employee)) DESC
3	-> HashAggregate (cost=305.85..338.35 rows=867 width=45)
4	Group Key: p.id_project
5	Filter: (count(ts.employee) > 3)
6	-> Hash Right Join (cost=208.25..288.35 rows=3500 width=41)
7	Hash Cond: (t.id_project = p.id_project)
8	-> Hash Right Join (cost=101.75..172.65 rows=3500 width=8)
9	Hash Cond: (ts.id_task = t.id_task)
10	-> Seq Scan on task_steps ts (cost=0.00..63.01 rows=3001 width=8)
11	-> Hash (cost=58.00..58.00 rows=3500 width=8)
12	-> Seq Scan on task t (cost=0.00..58.00 rows=3500 width=8)
13	-> Hash (cost=74.00..74.00 rows=2600 width=37)
14	-> Seq Scan on project p (cost=0.00..74.00 rows=2600 width=37)

Successfully run. Total query runtime: 92 msec. 14 rows affected. X

Total rows: 14 of 14 Query complete 00:00:00.092 Ln 2, Col 1

Рис. 20 – INDEX. Седьмой запрос. План без индексации

Query Query History Data Output Messages Notifications

```

1 create index project_index on project(id_project);
2
3 explain
4 select p.id_project, p.project_name, count(ts.employee)
5 from project as p
6 left outer join task as t on p.id_project = t.id_project
7 left outer join task_steps as ts on ts.id_task = t.id_task
8 left outer join schedule on schedule.id_schedule = ts.id_schedule
9 group by p.id_project
10 having count(ts.employee) > 3
11 order by employee_number desc
12

```

QUERY PLAN

1	Sort (cost=379.53..381.68 rows=860 width=45)
2	Sort Key: (count(ts.employee)) DESC
3	-> HashAggregate (cost=305.37..337.61 rows=860 width=45)
4	Group Key: p.id_project
5	Filter: (count(ts.employee) > 3)
6	-> Hash Right Join (cost=207.78..287.87 rows=3500 width=41)
7	Hash Cond: (t.id_project = p.id_project)
8	-> Hash Right Join (cost=101.75..172.65 rows=3500 width=8)
9	Hash Cond: (ts.id_task = t.id_task)
10	-> Seq Scan on task_steps ts (cost=0.00..63.01 rows=3001 width=8)
11	-> Hash (cost=58.00..58.00 rows=3500 width=8)
12	-> Seq Scan on task t (cost=0.00..58.00 rows=3500 width=8)
13	-> Hash (cost=73.79..73.79 rows=2579 width=37)
14	-> Seq Scan on project p (cost=0.00..73.79 rows=2579 width=37)

Successfully run. Total query runtime: 36 msec. 14 rows affected. X

Total rows: 14 of 14 Query complete 00:00:00.036 Ln 1, Col 51

Рис. 21 – INDEX. Седьмой запрос. План с простым индексом

Query Query History Data Output Messages Notifications

```

1 create index project_index on project(id_project);
2 create index project_name_index on project(project_name);
3
4 explain
5 select p.id_project, p.project_name, count(ts.employee)
6 from project as p
7 left outer join task as t on p.id_project = t.id_project
8 left outer join task_steps as ts on ts.id_task = t.id_task
9 left outer join schedule on schedule.id_schedule = ts.id_schedule
10 group by p.id_project
11 having count(ts.employee) > 3
12 order by employee_number desc
13

```

QUERY PLAN

- Sort (cost=379.53..381.68 rows=860 width=45)
- Sort Key: (count(ts.employee)) DESC
- HashAggregate (cost=305.37..337.61 rows=860 width=45)
- Group Key: p.id_project
- Filter: (count(ts.employee) > 3)
- Hash Right Join (cost=207.78..287.87 rows=3500 width=41)
- Hash Cond: (t.id_project = p.id_project)
- Hash Right Join (cost=101.75..172.65 rows=3500 width=8)
- Hash Cond: (ts.id_task = t.id_task)
- Seq Scan on task_steps ts (cost=0.00..63.01 rows=3001 width=8)
- Hash (cost=58.00..58.00 rows=3500 width=8)
- Seq Scan on task t (cost=0.00..58.00 rows=3500 width=8)
- Hash (cost=73.79..73.79 rows=2579 width=37)
- Seq Scan on project p (cost=0.00..73.79 rows=2579 width=37)

Successfully run. Total query runtime: 30 msec. 14 rows affected.

Total rows: 14 of 14 Query complete 00:00:00.030 Ln 1, Col 1

Рис. 22 – INDEX. Седьмой запрос. План с составным индексом 1

Query Query History Data Output Messages Notifications

```

1 create index project_index on project(id_project);
2 create index project_name_index on project(project_name);
3 create index task_index on task(id_task);
4
5 explain
6 select p.id_project, p.project_name, count(ts.employee)
7 from project as p
8 left outer join task as t on p.id_project = t.id_project
9 left outer join task_steps as ts on ts.id_task = t.id_task
10 left outer join schedule on schedule.id_schedule = ts.id_schedule
11 group by p.id_project
12 having count(ts.employee) > 3
13 order by employee_number desc
14

```

QUERY PLAN

- Sort (cost=379.53..381.68 rows=860 width=45)
- Sort Key: (count(ts.employee)) DESC
- HashAggregate (cost=305.37..337.61 rows=860 width=45)
- Group Key: p.id_project
- Filter: (count(ts.employee) > 3)
- Hash Right Join (cost=207.78..287.87 rows=3500 width=41)
- Hash Cond: (t.id_project = p.id_project)
- Hash Right Join (cost=101.75..172.65 rows=3500 width=8)
- Hash Cond: (ts.id_task = t.id_task)
- Seq Scan on task_steps ts (cost=0.00..63.01 rows=3001 width=8)
- Hash (cost=58.00..58.00 rows=3500 width=8)
- Seq Scan on task t (cost=0.00..58.00 rows=3500 width=8)
- Hash (cost=73.79..73.79 rows=2579 width=37)
- Seq Scan on project p (cost=0.00..73.79 rows=2579 width=37)

Successfully run. Total query runtime: 79 msec. 14 rows affected.

Total rows: 14 of 14 Query complete 00:00:00.079 Ln 3, Col 24

Рис. 23 – INDEX. Седьмой запрос. План с составным индексом 2

Выводы:

...И вот именно на этом этапе работы выяснилось, что в таблицу этапов заданий, собственно, не хватает самого важного: столбца с сотрудниками, над этими этапами работающими. Пришлось в уже прописанную базу данных, временно отключая часть проверок, добавлять столбец с информацией. Но — все-таки — база побеждена и дополнена.

Также всплыло много занятных артефактов от заполнения таблицы данными. Заполнялись они кодом, не вручную, поэтому, например, в списке отделов у меня обнаружилось штук пять отделов бухгалтерии. Ну, видимо, таков путь для этой организации — существовать с таким количеством бумажек.

Поэтому, собственно, в третьем задании в запросе для delete я использую именно название отдела. Чтоб охватить ВСЕ отделы ИТ, которых, так уж получилось, тоже создалось несколько.

Про долгосрочное планирование и постановку дедлайнов на 2090-е годы я молчу. Тут уж просто придется считать, что этой организации настолько долгоиграющие контракты.

Очень наглядно можно увидеть, как изменяется время обработки запроса при удачно и не очень составленной составной индексации. Это особенно видно по седьмому запросу — на рисунках 22 и 23. Пара ключей, которые идут в паре, уменьшают время обработки на 0,062 (по сравнению с временем обработки запроса, выполненного вообще без индексации). Но стоит добавить не тот индекс — и сразу 0,030 превращается в 0,079. И как-то этим фактом даже приводит в восторг.