

**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

**«Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»**

Выполнил:

студент группы К32402

Пластун Елизавета Олеговна

(подпись)

Проверил(а):

Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург

2023 г.

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.

Вариант 13. БД «Ресторан»

Описание предметной области: Необходимо создать систему для обслуживания заказов клиентов в ресторане.

Сотрудники ресторана – повара и официанты.

За каждым официантом закреплены определенные столы за смену. Клиенты могут бронировать столы заранее.

Каждый повар может готовить определенный набор блюд.

Официант принимает заказ от стола и передает его на кухню. Шеф-повар распределяет блюда для приготовления между поварами. В одном заказе может быть несколько одинаковых или разных блюд.

Запас продуктов на складе не должен быть ниже заданного значения.

Цена заказа складывается из стоимости ингредиентов и наценки, которая составляет 40% стоимости ингредиентов.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Паспортные данные сотрудника. Категория сотрудника. Должность сотрудника. Оклад сотрудника. Наименование ингредиента. Код ингредиента. Дата закупки. Объем закупки. Количество продукта на складе. Необходимый запас продукта. Срок годности. Цена ингредиента. Калорийность (на 100г продукта). Поставщик. Наименование блюда. Код блюда. Объем ингредиента. Номер стола. Дата заказа. Код заказа. Количество. Название блюда. Ингредиенты, входящие в блюдо. Тип ингредиента.

Задание 1.1 (ЛР 1 БД). Выполните инфологическое моделирование базы данных системы. (Ограничения задать самостоятельно.)

Задание 1.2. Создайте логическую модель БД, используя ИЛМ (задание 1.1). Используйте необходимые средства поддержки целостности данных в СУБД.

Задание 2. Создать запросы:

- Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.
- Рассчитать премию каждого официанта за последний месяц (5% от стоимости каждого заказа).
- Подсчитать, сколько ингредиентов содержит каждое блюдо.
- Вывести название блюда, содержащее максимальное число ингредиентов.
- Какой повар может приготовить максимальное число видов блюд?
- Сколько закреплено столов за каждым из официантов?
- Какой из ингредиентов используется в максимальном количестве блюд?

Задание 3. Создать представление:

- для расчета стоимости ингредиентов для заданного блюда;
- количество приготовленных блюд по каждому блюду за определенную дату.

Выполнение:

Запросы:

- Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.

Query

Query History

```
1 select * from (
2 select id_employee, sum(price) as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)
3 where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1 )group by id_employee)as d
4 join "lr1.2".employee using(id_employee)
5 where total_price = (
6 select max(b) from (
7 select sum(price) as b from "lr1.2".employee
8 join "lr1.2".order using (id_employee)
9 where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1 )group by id_employee) as c)
```

Data Output

Messages

Notifications

	id_employee [PK] integer	total_price double precision	full_name text	passport_data text	id_position integer
1	8	98000	овна	блорпвачстми	1

```

select * from (
select id_employee, sum(price) as total_price from "lr1.2".employee join
"lr1.2".order using (id_employee)
where (id_position = 1 and (extract(month from now()) - extract (month from
date_order)) = 1 )group by id_employee)as d
join "lr1.2".employee using(id_employee)
where total_price = (
select max(b) from (
select sum(price) as b from "lr1.2".employee
join "lr1.2".order using (id_employee)
where (id_position = 1 and (extract(month from now()) - extract (month from
date_order)) = 1 )group by id_employee) as c)

```

- Рассчитать премию каждого официанта за последний месяц (5% от стоимости каждого заказа).

Query Query History

```

1 select id_employee, (total_price*0.05)as present from(select id_employee, sum(price)
2 as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)
3 where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1 )
4 group by id_employee)as d
5 join "lr1.2".employee using(id_employee)
6
7

```

Data Output Messages Notifications

	id_employee [PK] integer	present double precision
1	6	7.3500000000000005
2	7	4.9
3	8	4900

```

select id_employee, (total_price*0.05)as present from(select id_employee,
sum(price)
as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)
where (id_position = 1 and (extract(month from now()) - extract (month from
date_order)) = 1 )
group by id_employee)as d
join "lr1.2".employee using(id_employee)

```

- Подсчитать, сколько ингредиентов содержит каждое блюдо.

Query Query History

```

1 select id_dish, name_dish, amount from (select id_dish, count(id_ingredient) as amount
2 from "lr1.2".dish_list
3 group by id_dish) as d join "lr1.2".dish using (id_dish) order by id_dish
4
5

```

Data Output Messages Notifications

	id_dish integer	name_dish text	amount bigint
1	1	сопли	3
2	2	кирпичи	2
3	3	яйца	1
4	4	паша	1

```

select id_dish, name_dish, amount from (select id_dish, count(id_ingredient)
as amount
from "lr1.2".dish_list
group by id_dish) as d join "lr1.2".dish using (id_dish) order by id_dish

```

- Вывести название блюда, содержащее максимальное число ингредиентов.

Query Query History

```

1
2 select id_dish, name_dish, amount from(select id_dish, count(id_ingredient) as amount
3 from "lr1.2".dish_list
4 group by id_dish) as d join "lr1.2".dish using (id_dish) where amount =
5 (select max(amount) from(select count(id_ingredient) as amount
6 from "lr1.2".dish_list group by id_dish) as f)
7

```

Data Output Messages Notifications

	id_dish integer	name_dish text	amount bigint
1	1	сопли	3

```

select id_dish, name_dish, amount from(select id_dish, count(id_ingredient)
as amount
from "lr1.2".dish_list
group by id_dish) as d join "lr1.2".dish using (id_dish) where amount =
(select max(amount) from(select count(id_ingredient) as amount
from "lr1.2".dish_list group by id_dish) as f)

```

- Какой повар может приготовить максимальное число видов блюд?

```
select id_employee, full_name, id_dish from
(select id_employee, count(id_dish) as amount_dish
from "lr1.2".dish
group by id_employee) as d join "lr1.2".dish using(id_employee)
join "lr1.2".employee using(id_employee) where amount_dish =
(select max(amount_dish) from(select count(id_dish) as amount_dish
from "lr1.2".dish group by id_employee) as f)
```

Data Output Messages Notifications			
	id_employee integer	full_name text	id_dish integer
1	3	шрукашпукла	1
2	3	шрукашпукла	5
3	3	шрукашпукла	7

```
select id_employee, full_name, id_dish from
(select id_employee, count(id_dish) as amount_dish
from "lr1.2".dish
group by id_employee) as d join "lr1.2".dish using(id_employee)
join "lr1.2".employee using(id_employee) where amount_dish =
(select max(amount_dish) from(select count(id_dish) as amount_dish
from "lr1.2".dish group by id_employee) as f)
```

- Сколько закреплено столов за каждым из официантов?

Query Query History

```
1 select id_employee, full_name, tabless_count from
2 (select id_employee, count(id_table) as tabless_count from "lr1.2".table
3 group by id_employee) as d join "lr1.2".employee using(id_employee)
```

Data Output Messages Notifications

	id_employee integer	full_name text	tabless_count bigint
1	1	рашщрклдабуккы	3
2	8	овна	2
3	6	попейг	3
4	7	Марина Михайловна ♥ лучшая	2

```
select id_employee, full_name, tabless_count from
```

```
(select id_employee, count(id_table) as tabless_count from "lr1.2".table
group by id_employee) as d join "lr1.2".employee using(id_employee)
```

- Какой из ингредиентов используется в максимальном количестве блюд?

Query Query History

```

1 select id_ingredient, type_ingredient, amount_dish from
2 (select id_ingredient, count(id_dish) as amount_dish
3 from "lr1.2".dish_list
4 group by id_ingredient) as d join "lr1.2".ingredient using(id_ingredient)
5 where amount_dish =
6 (select max(amount_dish) from(select count(id_dish) as amount_dish
7 from "lr1.2".dish_list group by id_ingredient) as f)
8 |

```

Data Output Messages Notifications

	id_ingredient integer	type_ingredient text	amount_dish bigint
1	4	человечина	3

```
select id_ingredient, type_ingredient, amount_dish from
(select id_ingredient, count(id_dish) as amount_dish
from "lr1.2".dish_list
group by id_ingredient) as d join "lr1.2".ingredient using(id_ingredient)
where amount_dish =
(select max(amount_dish) from(select count(id_dish) as amount_dish
from "lr1.2".dish_list group by id_ingredient) as f)
```

Представления:

- для расчета стоимости ингредиентов для заданного блюда;

Query Query History

```

1 create or replace view price_for_ingredient as
2 select name_dish, sum(price_100_gramm) from (select id_dish from "lr1.2".dish
3 where name_dish = 'супли') as s
4 join "lr1.2".dish_list using(id_dish) join "lr1.2".ingredient using(id_ingredient)
5 join "lr1.2".dish using(id_dish) group by name_dish
6
7
8
9

```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 121 msec.

QueryQuery History

12

```
select * from price_for_ingredient
```

Data OutputMessagesNotifications

	name_dish text	sum bigint
1	сопли	60120

- количество приготовленных блюд по каждому блюду за определенную дату.

QueryQuery History

123456

```
create or replace view amount_dishes_for_date as
select id_dish,name_dish,sum_amount from
(select id_dish, sum(amount) as sum_amount from "lr1.2".order
join "lr1.2".order_list using(id_order) where date_order = '2023-04-05'
group by id_dish) as f join "lr1.2".dish using(id_dish)
```

Data OutputMessagesNotifications

CREATE VIEW

Query returned successfully in 42 msec.

create or replace view amount_dishes_for_date as


```
select id_dish,name_dish, sum_amount from
(select id_dish, sum(amount) as sum_amount from "lr1.2".order
join "lr1.2".order_list using(id_order) where date_order = '2023-04-05'
group by id_dish) as f join "lr1.2".dish using(id_dish)
```

Query

Query History

1

select * from amount_dishes_for_date

2

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	id_dish integer	name_dish text	sum_amount bigint
1	1	сопли	9
2	2	кирпичи	1
3	3	яйца	2

Модификация данных

```
INSERT INTO "lr1.2".employee(
    id_employee, full_name, passport_datas, id_position)
VALUES ((select max(id_employee)+1 from "lr1.2".employee),
        'паляхова жанна олеговна', '5617 676756 passport',
        (select id_position from "lr1.2".position where name_position =
'повар'));
```

Query

Query History

```

1 INSERT INTO "lr1.2".employee(
2     id_employee, full_name, passport_datas, id_position)
3     VALUES ((select max(id_employee)+1 from "lr1.2".employee),
4         'паляхова жанна олеговна', '5617 676756 passport',
5         (select id_position from "lr1.2".position where name_position = 'повар'));

```

Data Output

Messages

Notifications

INSERT 0 1

Query returned successfully in 123 msec.

```

UPDATE "lr1.2"."order"
SET price=price*5
WHERE id_order in ((select id_order from "lr1.2"."dish"
                    join "lr1.2"."dish_list" using(id_dish)
                    join "lr1.2"."order_list" using(id_dish)
                    where id_ingredient = 4));

```

	id_order [PK] integer	date_order date	date_booking date	id_employee integer	employment_status text	id_table integer	id_customer integer	price double precision
1	1	2023-05-05	2005-05-09	1	занят	1	1	390
2	2	2023-05-09	2005-05-09	1	холост	2	2	780
3	3	2023-05-16	2005-05-09	1	занят	3	3	87933
4	4	2023-04-05	2023-04-05	6	пгл	3	1	78
5	5	2023-04-05	2023-04-05	7	лло	2	1	90
6	6	2023-04-05	2023-04-05	8	занят	1	1	8000
7	7	2023-04-05	2023-04-05	6	занят	3	1	69
8	8	2023-04-05	2023-04-05	7	занят	1	1	8
9	9	2023-04-05	2023-04-05	8	занят	1	2	90000

Data Output

Messages

Notifications

	id_order [PK] integer	date_order date	date_booking date	id_employee integer	employment_status text	id_table integer	id_customer integer	price double precision
1	1	2023-05-05	2005-05-09	1	занят	1	1	1950
2	2	2023-05-09	2005-05-09	1	холост	2	2	3900
3	3	2023-05-16	2005-05-09	1	занят	3	3	87933
4	4	2023-04-05	2023-04-05	6	пгл	3	1	78
5	5	2023-04-05	2023-04-05	7	лло	2	1	450
6	6	2023-04-05	2023-04-05	8	занят	1	1	40000
7	7	2023-04-05	2023-04-05	6	занят	3	1	69
8	8	2023-04-05	2023-04-05	7	занят	1	1	8
9	9	2023-04-05	2023-04-05	8	занят	1	2	450000

Query Query History

```

1 UPDATE "lr1.2"."order"
2   SET price=price*5
3   WHERE id_order in ((select id_order from "lr1.2"."dish"
4                        join "lr1.2"."dish_list" using(id_dish)
5                        join "lr1.2"."order_list" using(id_dish)
6                        where id_ingredient = 4));

```

Data Output Messages Notifications

UPDATE 5

Query returned successfully in 38 msec.

```

DELETE FROM "lr1.2"."order_list"
  WHERE id_order = (select id_order from "lr1.2".order
                    where price = (select min(price) from «lr1.2".order));

```

Data Output Messages Notifications

	id_order integer	id_employee integer	id_dish integer	status text	comment text	amount integer	id_order_list [PK] integer
1	1	1	1	готово	без людей	4	1
2	2	2	2	не готово	без денег	7	2
3	3	3	3	не готово	без слез	8	3
4	1	2	1	готов	[null]	1	4
5	2	3	1	готов	[null]	1	5
6	1	2	2	готов	[null]	1	6
7	5	3	3	готов	[null]	1	7
8	9	1	1	готов	[null]	5	8
9	6	1	2	готов	[null]	1	9
10	9	1	3	готов	[null]	1	10
11	5	1	1	готов	[null]	4	11
12	1	1	3	готов	[null]	2	12
13	8	1	4	готово	aaaa	4	13

Query Query History

```
1 DELETE FROM "lr1.2"."order_list"
2 WHERE id_order = (select id_order from "lr1.2".order
3                     where price = (select min(price) from "lr1.2".order));
4
5
6
7
```

Data Output Messages Notifications

DELETE 1

	id_order integer	id_employee integer	id_dish integer	status text	comment text	amount integer	id_order_list [PK] integer
1	1	1	1	готово	без людей	4	1
2	2	2	2	не готово	без денег	7	2
3	3	3	3	не готово	без слез	8	3
4	1	2	1	готов	[null]	1	4
5	2	3	1	готов	[null]	1	5
6	1	2	2	готов	[null]	1	6
7	5	3	3	готов	[null]	1	7
8	9	1	1	готов	[null]	5	8
9	6	1	2	готов	[null]	1	9
10	9	1	3	готов	[null]	1	10
11	5	1	1	готов	[null]	4	11
12	1	1	3	готов	[null]	2	12

Создание индексов

QueryQuery History

1

select full_name from "lr1.2"."employee" where id_position = 1

2

3

4

Data OutputMessagesExplain XNotifications

full_name

text

1	рашщрклдабукыы
2	овна
3	попейг
4	Марина Михайловна 💖 лучшая

QueryQuery History

1

select full_name from "lr1.2"."employee" where id_position = 1

2

3

4

Data OutputMessagesExplain XNotifications

GraphicalAnalysisStatistics

employee

Query

Query History

Scratch Pad X

1

2

3

4

```
select full_name from "lr1.2"."employee" where id_position = 1
```

Data Output

Messages

Explain X

Notifications

Graphical

Analysis

Statistics

#	Node	Rows	
		Actual	Loops
1.	→ Seq Scan on employee as employee (rows=4 loops=1) Filter: (id_position = 1) Rows Removed by Filter: 8	4	1

Query

Query History

1

2

3

4

```
select full_name from "lr1.2"."employee" where id_position = 1
```

Data Output

Messages

Explain X

Notifications

Successfully run. Total query runtime: 40 msec.
1 rows affected.

Query

Query History

1

```
create index employee_name on "lr1.2"."employee"(id_position)
```

Data Output

Messages

Notifications

CREATE INDEX

Query returned successfully in 35 msec.

Query
Query History

1
2 `select * from pg_indexes where tablename = 'employee'`
3
4 `-- select full_name from "lr1.2"."employee" where id_position = 1`

Data Output
Messages
Explain
Notifications

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	lr1.2	employee	employee_pkey	[null]	CREATE UNIQUE INDEX employee_pkey ON "lr1.2".employee USING btree (id_employee)
2	lr1.2	employee	employee_passport_datas_passport_datas1_key	[null]	CREATE UNIQUE INDEX employee_passport_datas_passport_datas1_key ON "lr1.2".employee
3	lr1.2	employee	employee_name	[null]	CREATE INDEX employee_name ON "lr1.2".employee USING btree (id_position)

Query
Query History

1
2 `-- select * from pg_indexes where tablename = 'employee'`
3
4 `select full_name from "lr1.2"."employee" where id_position = 1`

Data Output
Messages
Explain
Notifications

Successfully run. Total query runtime: 36 msec.

1 rows affected.

Стало быстрее на 4 мс

Query
Query History

1
2 `select id_employee, (total_price*0.05)as present from(select id_employee, sum(price)`
3 `as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)`
4 `where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1)`
5 `group by id_employee)as d`
6 `join "lr1.2".employee using(id_employee)`

Data Output
Messages
Explain
Notifications

	id_employee [PK] integer	present double precision
1	8	24500
2	6	7.3500000000000005
3	7	22.900000000000002

Query
Query History

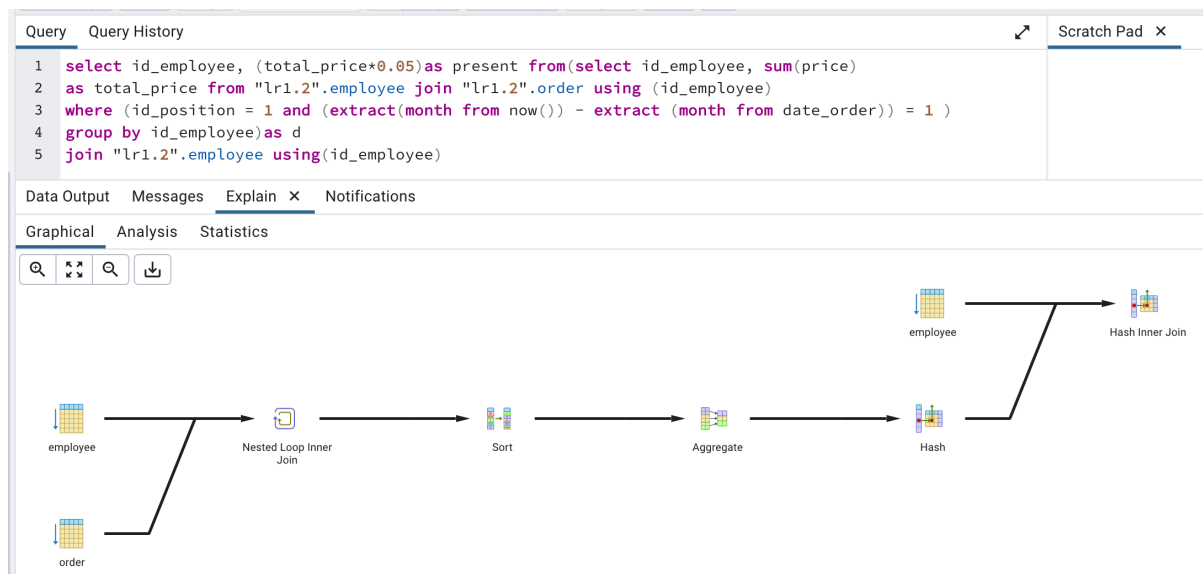
1
2 `select id_employee, (total_price*0.05)as present from(select id_employee, sum(price)`
3 `as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)`
4 `where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1)`
5 `group by id_employee)as d`
6 `join "lr1.2".employee using(id_employee)`

Data Output
Messages
Explain
Notifications

Successfully run. Total query runtime: 47 msec.

3 rows affected.

Query		Query History	Scratch Pad	
1		<code>select id_employee, (total_price*0.05)as present from(select id_employee, sum(price)</code>		
2		<code>as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)</code>		
3		<code>where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1)</code>		
4		<code>group by id_employee)as d</code>		
5		<code>join "lr1.2".employee using(id_employee)</code>		
Data Output		Messages	Explain	Notifications
Graphical		Analysis	Statistics	
#	Node	Rows Actual	Loops	
1.	→ Hash Inner Join (rows=3 loops=1) Hash Cond: (employee.id_employee = employee_1.id_employee)		3	1
2.	→ Seq Scan on employee as employee (rows=12 loops=1)		12	1
3.	→ Hash (rows=3 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB		3	1
4.	→ Aggregate (rows=3 loops=1)		3	1
5.	→ Sort (rows=6 loops=1)		6	1
6.	→ Nested Loop Inner Join (rows=6 loops=1) Join Filter: (employee_1.id_employee = "order".id_employee)		6	1
7.	→ Seq Scan on employee as employee_1 (rows=4 loops=1) Filter: (id_position = 1) Rows Removed by Filter: 8		4	1
8.	→ Seq Scan on order as order (rows=6 loops=4) Filter: ((EXTRACT(month FROM now()) - EXTRACT(month FROM date_order)) = '1':numeric) Rows Removed by Filter: 3		6	4



```

6
7 create unique index bonus_at_work on "lr1.2".employee(id_employee, id_position)

```

Data Output Messages Explain X Notifications

CREATE INDEX

Query returned successfully in 41 msec.

Query
Query History

```

1 select id_employee, (total_price*0.05)as present from(select id_employee, sum(price)
2 as total_price from "lr1.2".employee join "lr1.2".order using (id_employee)
3 where (id_position = 1 and (extract(month from now()) - extract (month from date_order)) = 1 )
4 group by id_employee)as d
5 join "lr1.2".employee using(id_employee)
6
7 -- create unique index bonus_at_work on "lr1.2".employee(id_employee, id_position)

```

Data Output
Messages
Explain
Notifications

Successfully run. Total query runtime: 42 msec.
3 rows affected.

Data Output
Messages
Explain
Notifications

Graphical
Analysis
Statistics

#	Node	Rows Actual	Loops
1.	→ Hash Inner Join (rows=3 loops=1) Hash Cond: (employee.id_employee = employee_1.id_employee)	3	1
2.	→ Seq Scan on employee as employee (rows=12 loops=1)	12	1
3.	→ Hash (rows=3 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	3	1
4.	→ Aggregate (rows=3 loops=1)	3	1
5.	→ Sort (rows=6 loops=1)	6	1
6.	→ Nested Loop Inner Join (rows=6 loops=1) Join Filter: (employee_1.id_employee = "order".id_employee)	6	1
7.	→ Seq Scan on employee as employee_1 (rows=4 loops=1) Filter: (id_position = 1) Rows Removed by Filter: 8	4	1
8.	→ Seq Scan on order as order (rows=6 loops=4) Filter: ((EXTRACT(month FROM now()) - EXTRACT(month FROM date_order)) = '1':numeric) Rows Removed by Filter: 3	6	4

Стало быстрее на 5 мс

Query
Query History

```

1 drop index "lr1.2".employee_name

```

Data Output
Messages
Explain
Notifications

DROP INDEX

Query returned successfully in 59 msec.

Query		Query History
1	<code>drop index "lr1.2".bonus_at_work</code>	
Data Output		Messages
DROP INDEX		Query returned successfully in 49 msec.

индексы помогают сократить время сложного запроса, но на примере простых запросов мы

видим что планировщик считает что лучше просканировать обычным способом

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 примерно столько же. Это связано с небольшим количеством данных в таблице.