

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий

Лабораторная работа №2.2
«Запросы на выборку и модификацию
данных, представления и индексы в
PostgreSQL»
по дисциплине:
«Базы данных»

Выполнила:
студентка II курса ИКТ
группы К3242
Липина Ольга Андреевна

Проверил:
Говорова Марина Михайловна

Санкт-Петербург

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Программное обеспечение: СУБД PostgreSQL 1X, 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.

Ход работы:

Часть 1. Запросы

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

```

WITH Temp_2 (id, total_price)
AS(
WITH Temp_ (id_worker, full_name, id_order, price_order)
AS (
SELECT workers.id, full_name, id_order, SUM(order_content.quantity * price * ingr_quantity)
FROM "Restaraunt_schema".workers
INNER JOIN "Restaraunt_schema".orders
ON "Restaraunt_schema".workers.id="Restaraunt_schema".orders.worker_id
INNER JOIN "Restaraunt_schema".order_content
ON "Restaraunt_schema".orders.id="Restaraunt_schema".order_content.id_order
INNER JOIN "Restaraunt_schema".dishes
ON "Restaraunt_schema".order_content.id_dish="Restaraunt_schema".dishes.id
INNER JOIN "Restaraunt_schema".ingr_quan
ON "Restaraunt_schema".dishes.id="Restaraunt_schema".ingr_quan.id_dish
INNER JOIN "Restaraunt_schema".purch_ingr
ON "Restaraunt_schema".ingr_quan.id_ingr="Restaraunt_schema".purch_ingr.id_ingr
WHERE extract(MONTH FROM "Restaraunt_schema".orders.date)=3
AND ("Restaraunt_schema".orders.status='paid')
GROUP BY workers.id, id_order
)
)
SELECT id_worker, SUM(price_order) FROM Temp_
GROUP BY id_worker
)
SELECT * FROM Temp_2
INNER JOIN "Restaraunt_schema".workers USING(id)
WHERE total_price=(SELECT MAX(total_price) FROM Temp_2)

```

Data Output								Explain	Messages	Notifications
id [PK] integer	total_price numeric	full_name character varying	category character varying	passport_data character varying	position character varying	salary numeric				
1	25	52314.800	Штофф Фелиция Ко...	team_leader	1922 578361	waiter	2257			

Рисунок 1 – Скриншот результата запроса 1

Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа).

```
WITH premium (full_name, premium)
AS (
WITH price_dish (id_dish, name_dish, price_dish)
AS (
    SELECT "Restaraunt_schema".dishes.id,
    "Restaraunt_schema".dishes.name,
    SUM(price*ingr_quantity)
    FROM "Restaraunt_schema".dishes
    INNER JOIN "Restaraunt_schema".ingr_quan
    ON "Restaraunt_schema".dishes.id="Restaraunt_schema".ingr_quan.id_dish
    INNER JOIN "Restaraunt_schema".purch_ingr
    ON "Restaraunt_schema".ingr_quan.id_ingr="Restaraunt_schema".purch_ingr.id_ingr
    group by "Restaraunt_schema".dishes.id
    )
SELECT full_name, SUM(quantity) * SUM(price_dish.id_dish*price_dish)*0.05 AS premium
FROM "Restaraunt_schema".workers
INNER JOIN "Restaraunt_schema".orders
ON "Restaraunt_schema".workers.id="Restaraunt_schema".orders.worker_id
INNER JOIN "Restaraunt_schema".order_content
ON "Restaraunt_schema".orders.id="Restaraunt_schema".order_content.id_order
INNER JOIN price_dish
ON "Restaraunt_schema".order_content.id_dish=price_dish.id_dish
WHERE date > current_date - interval '10' day AND "Restaraunt_schema".orders.status='paid'
group by full_name
)
select * FROM premium
UNION
SELECT full_name, 0 FROM "Restaraunt_schema".workers
WHERE workers.position='waiter' AND workers.full_name not in (SELECT full_name from premium)
```

	Data Output	Explain	Messages	Notifications
	<div> <div>full_name</div> <div>character varying</div> </div>		<div> <div>premium</div> <div>numeric</div> </div>	
1	Комсомольцева Карина Даниловна		0	
2	Данилова Марья Даниловна		0	
3	Комсомольцева Виктория Даниловна		186.000	
4	Штофф Модестас Царевич		0	
5	Воедотов Енисей Королевич		0	
6	Горнова Карина Романовна		0	
7	Штофф Фелиция Королевна		25720.24000	

Рисунок 2 – Скриншот результата запроса 2

Подсчитать, сколько ингредиентов содержит каждое блюдо.

```
SELECT dishes.name, COUNT(ingr_quan.id_ingr) FROM
"Restaraunt_schema".dishes INNER JOIN
"Restaraunt_schema".ingr_quan ON dishes.id =
ingr_quan.id_dish GROUP BY dishes.id;
```

	Data Output	Explain	Messages	I
	<div> <div>name</div> <div>character varying</div> </div>		<div> <div>count</div> <div>bigint</div> </div>	
1	cake		2	
2	boiled pork		2	
3	pasta		2	
4	ice cream		2	
5	salad		2	
6	smuzy		2	
7	meat chowder		3	

Рисунок 3 – Скриншот результата запроса 3

Вывести название блюда, содержащее максимальное число ингредиентов. содержит каждое блюдо.

```
WITH how_many (name, quantity)
AS (
SELECT dishes.name, COUNT(ingr_quan.id_ingr)
      FROM "Restaunt_schema".dishes
     INNER JOIN "Restaunt_schema".ingr_quan ON dishes.id = ingr_quan.id_dish
    GROUP BY dishes.id
)
SELECT * FROM how_many WHERE quantity = (SELECT MAX(quantity) FROM how_many)
```

	Data Output	Explain	Messages	Notifications
	name character varying	quantity bigint		
1	meat chowder	3		

Рисунок 4 – Скриншот результата запроса 4

Какой повар может приготовить максимальное число видов блюд?

```
WITH how_many (name, quantity) AS ( SELECT workers.full_name, COUNT(worker_dish.dish_id) FROM
"Restaunt_schema".workers INNER JOIN "Restaunt_schema".worker_dish ON workers.id =
worker_dish.worker_id GROUP BY workers.id ) SELECT * FROM how_many WHERE quantity = (SELECT
MAX(quantity) FROM how_many)
```

	Data Output	Explain	Messages	Notifica
	name character varying	quantity bigint		
1	Горнова Марья Да...	3		

Рисунок 5 – Скриншот результата запроса 5

Сколько закреплено столов за каждым из официантов?

```
SELECT workers.full_name, COUNT(orders.table_id)
      FROM "Restaraunt_schema".workers
      INNER JOIN "Restaraunt_schema".orders ON workers.id =
orders.worker_id
      GROUP BY workers.id
```

	Data Output	Explain	Messages	Notifications
	full_name character varying			count bigint
1	Штофф Фелиция Королевна			3
2	Горнова Карина Романовна			6
3	Комсомольцева Карина Даниловна			4
4	Воедотов Енисей Королевич			4
5	Комсомольцева Виктория Даниловна			1

Рисунок 6 – Скриншот результата запроса 6

```
WITH how_many (name, quantity) AS ( SELECT ingredients.name,
COUNT(dishes.id) FROM "Restaunt_schema".dishes INNER JOIN
"Restaunt_schema".ingr_quan ON dishes.id = ingr_quan.id_dish INNER JOIN
"Restaunt_schema".ingredients ON ingr_quan.id_ingr = ingredients.id GROUP
BY ingredients.id ) SELECT * FROM how_many WHERE quantity = (SELECT
MAX(quantity) FROM how_many)
```

Data Output		Explain	Messages	N
	<div><div><div></div><div><div>name</div><div>character varying</div></div></div><div><div>quantity</div><div>bigint</div></div></div>			
1	salt			3
2	meat			3

Рисунок 7 – Скриншот результата запроса 7

Представления

для расчета стоимости ингредиентов для заданного блюда;

```
SELECT DISTINCT(dishes.name), ingredients.name, (ingr_quan.ingr_quantity * purch_ingr.price) AS
price FROM "Restaunt_schema".dishes INNER JOIN "Restaunt_schema".ingr_quan ON dishes.id =
ingr_quan.id_dish INNER JOIN "Restaunt_schema".ingredients ON ingr_quan.id_ingr =
ingredients.id INNER JOIN "Restaunt_schema".purch_ingr ON ingr_quan.id_ingr =
purch_ingr.id_ingr WHERE dishes.name='smuzy'
```

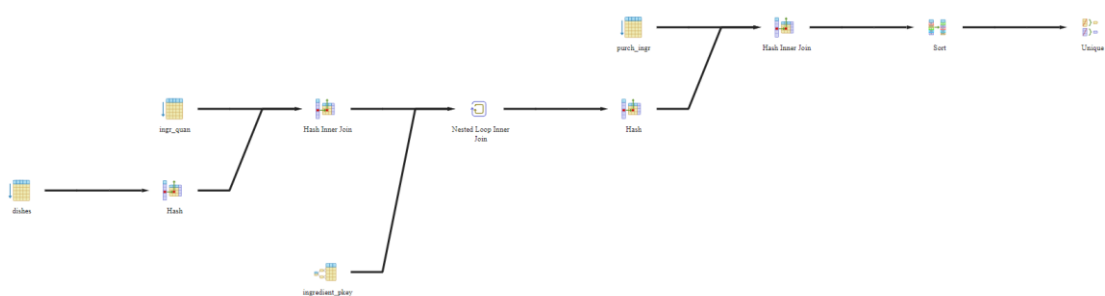


Рисунок 8 – SVG представление запроса 8

для всех поваров количество приготовленных блюд по каждому блюду за определенную дату.

```
SELECT workers.full_name, id_dish, SUM(quantity) FROM "Restaraunt_schema".orders INNER JOIN  
"Restaraunt_schema".order_content ON orders.id = order_content.id_order INNER JOIN  
"Restaraunt_schema".worker_dish ON order_content.id_dish = worker_dish.dish_id INNER JOIN  
"Restaraunt_schema".workers ON worker_dish.worker_id = workers.id WHERE date = '2022-03-01'  
GROUP BY workers.id, id_dish
```

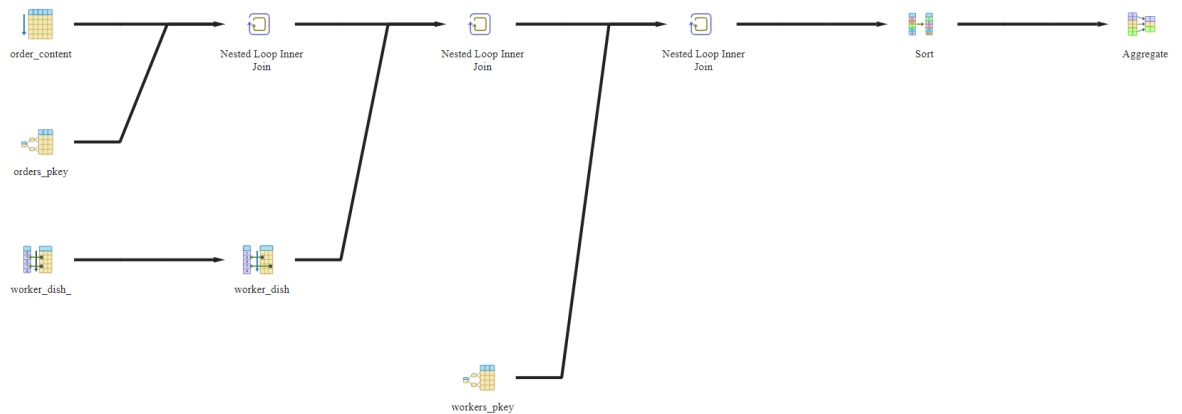


Рисунок 9 – SVG представление запроса 9

Часть 2. Запросы на модификацию данных

UPDATE

```
UPDATE "Restaraunt_schema".workers  
SET salary=(SELECT (salary*1.05) FROM "Restaraunt_schema".workers WHERE id=1)  
WHERE id = 1;
```

INSERT с подзапросами

```
INSERT INTO "Restaraunt_schema".workers(  
full_name, category, passport_data, "position", salary)  
VALUES ('Катанов Егор Вальеревич', 'intern', '1000 9876', 'cooker',  
(SELECT ROUND(AVG(salary),1) FROM "Restaraunt_schema".workers WHERE category='inte  
rn'));
```

DELETE

```
DELETE FROM "Restaraunt_schema".workers  
WHERE id = (SELECT MAX(id) FROM "Restaraunt_schema".workers);
```

```
1 INSERT INTO "Restaraunt_schema".workers(  
2     full_name, category, passport_data, "position", salary)  
3     VALUES ('Катанов Егор Вальеревич', 'intern', '1000 9876', 'cooker',  
4         (SELECT ROUND(AVG(salary),1) FROM "Restaraunt_schema".workers WHERE category='intern'));
```

Data Output Explain Messages Notifications

INSERT 0 1

Query returned successfully in 250 msec.

Рисунок 10 – INSERT запрос

```
1 DELETE FROM "Restaraunt_schema".workers  
2     WHERE id = (SELECT MAX(id) FROM "Restaraunt_schema".workers);
```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 104 msec.

Рисунок 11 – DELETE запрос

```

1 UPDATE "Restaunt_schema".workers
2     SET salary=(SELECT (salary*1.05) FROM "Restaunt_schema".workers WHERE id=1)
3     WHERE id = 1;

```

Data Output Explain Messages Notifications

UPDATE 1

Query returned successfully in 81 msec.

Рисунок 12 – UPDATE запрос

Часть 3. Простой и составной индекс

Copy Copy to Query Editor

```

SELECT full_name FROM "Restaunt_schema".workers
WHERE id IN (SELECT worker_id AS quant FROM "Restaunt_schema".
             GROUP BY worker_id
             HAVING COUNT(dish_id) > 2)

```

Messages

Successfully run. Total query runtime: 101 msec.
1 rows affected.

Рисунок 13 – С составным индексом

06.04.2022 23:07:04
Date

1
Rows Affected

135 msec
Duration

Copy

Copy to Query Editor

```
SELECT full_name FROM "Restaunt_schema".workers
WHERE id IN (SELECT worker_id AS quant FROM "Restaunt_schema".
             GROUP BY worker_id
             HAVING COUNT(dish_id) > 2)
```

Messages

Successfully run. Total query runtime: 135 msec.
1 rows affected.

Рисунок 14 – Без составного индекса

```
1 CREATE INDEX id_work ON "Restaunt_schema".workers(id)
```

Data Output Explain Messages Notifications

CREATE INDEX

Query returned successfully in 233 msec.

Рисунок 15 – Создание индекса

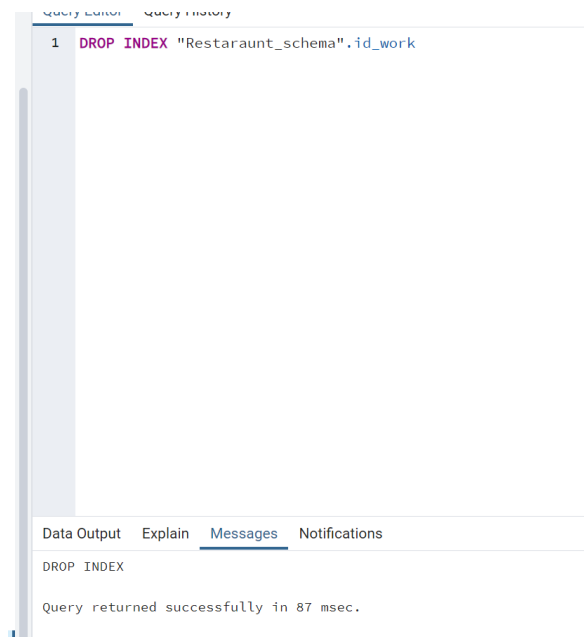


Рисунок 16 – Удаление индекса

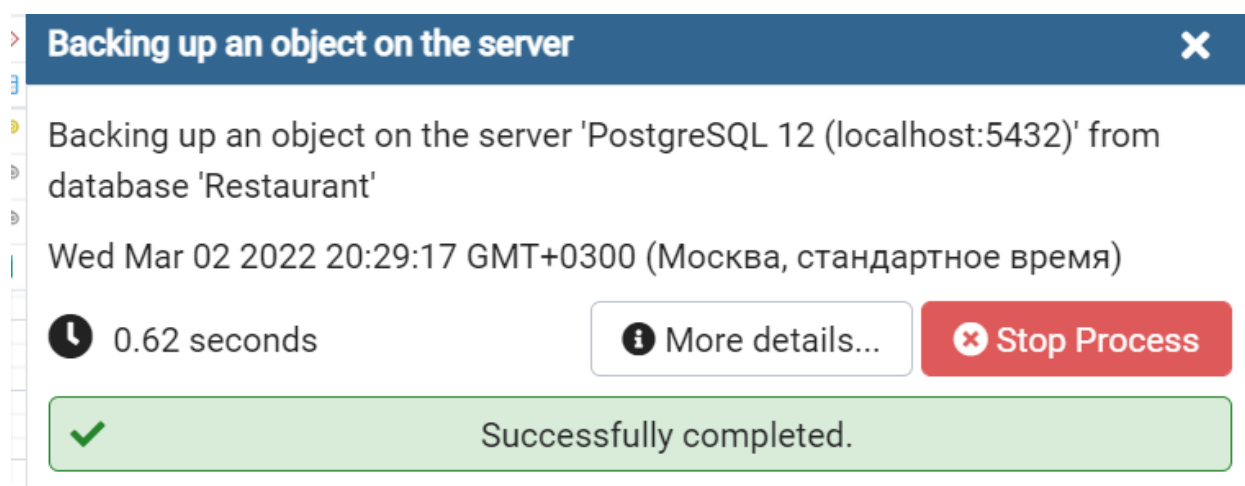


Рисунок 3 – Скриншот успешного создания дампа БД

С помощью созданного бэкапа база была восстановлена.

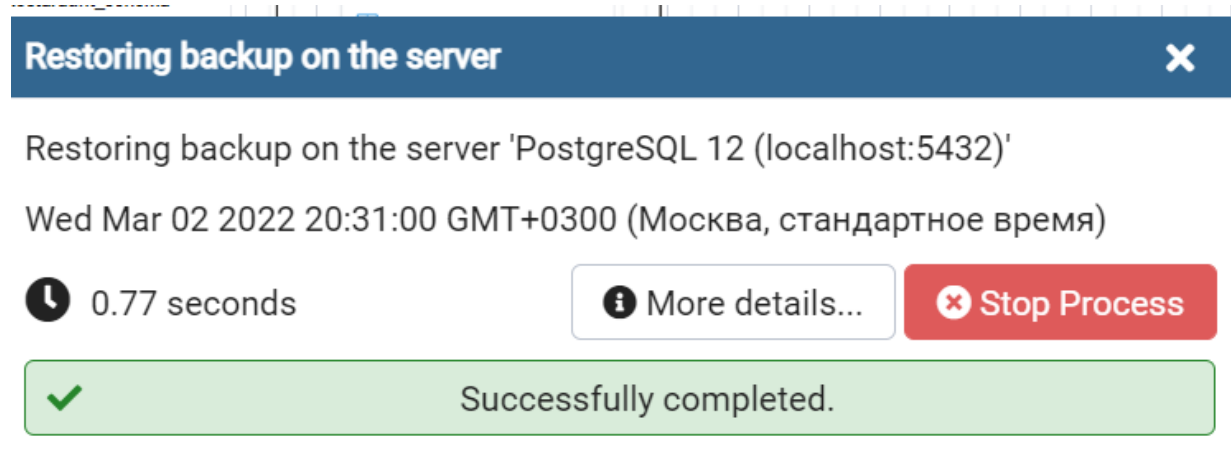


Рисунок 3 – Скриншот успешного восстановления БД

Выводы:

Выводы: с помощью pgadmin были созданы запросы на вывод, добавление, обновление и удаление данных с применением технологий объединения таблиц, подзапросов, обобщённых табличных выражение и т.п. Сравнила обработку двух запросов – с составными индексами запросы выполняются быстрее.