

Национальный исследовательский Университет ИТМО
Мегафакультет информационных и трансляционных технологий
Факультет мобильных и сетевых технологий

Проектирование и реализация баз данных

Лабораторная работа №4

Запросы на выборку и модификацию данных. Представления. Работа с индексами

Работу выполнил:

Фадеев Д.А.

Группа: К3239

Преподаватель:

Говорова М.М.

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.

Практическое задание

- Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
- Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
- Изучить графическое представление запросов и просмотреть историю запросов.
- 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 9. Оптовая База

Запросы к базе данных

- Вывести список поставщиков, которые поставляют все товары.

The screenshot shows a PostgreSQL query editor with a query window and a data output window.

Query:

```
1 SELECT slt2.supplier_id FROM(
2     SELECT slt1.supplier_id, COUNT(slt1.product_id) AS products_count FROM(
3         SELECT DISTINCT s.supplier_id, pr.product_id
4         FROM wholesale_base.supplier s
5         JOIN wholesale_base.purchase pu ON s.supplier_id = pu.supplier_id
6         JOIN wholesale_base.purchase_list pl ON pu.purchase_id = pl.purchase_id
7         JOIN wholesale_base.product pr ON pr.product_id= pl.product_id
8         ORDER BY s.supplier_id, pr.product_id
9     ) slt1
10    GROUP BY slt1.supplier_id
11 ) slt2
12 WHERE slt2.products_count = (SELECT COUNT(*) FROM wholesale_base.product)
```

Data Output:

supplier_id [PK] integer
1
6

- Определить поставщика, который поставляет каждый из товаров по самой низкой цене.

Query










Query History

```
1 SELECT slt.supplier_id, slt.product_id, slt.price
2 FROM(
3     SELECT s.supplier_id, pr.product_id, pl.price,
4     RANK() OVER (PARTITION BY pr.product_id ORDER BY pl.price) AS rnk
5     FROM wholesale_base.supplier s
6     JOIN wholesale_base.purchase pu ON s.supplier_id = pu.supplier_id
7     JOIN wholesale_base.purchase_list pl ON pu.purchase_id = pl.purchase_id
8     JOIN wholesale_base.product pr ON pr.product_id= pl.product_id
9     GROUP BY s.supplier_id, pr.product_id, pl.price
10    ORDER BY pr.product_id, pl.price, s.supplier_id
11 ) slt
12 WHERE slt.rnk = 1
13 GROUP BY slt.supplier_id, slt.product_id, slt.price, slt.rnk
14 ORDER BY slt.product_id, slt.price;
```

Data Output

Messages

Notifications



	supplier_id integer	product_id integer	price integer
1	7	1	118
2	10	2	101
3	5	3	114
4	6	4	234
5	4	5	364
6	8	6	115
7	8	7	222
8	6	8	123
9	6	9	234

- Чему равен общий суточный доход оптового склада за прошедший день?

Query

Query History

```
1 SELECT date_day, SUM(income-expenses) AS profit FROM(
2     SELECT o.date_of_sale AS date_day, SUM(ol.count * ol.price) AS income, 0 AS expenses
3     FROM wholesale_base.order o
4     JOIN wholesale_base.order_list ol ON o.order_id = ol.order_id
5     GROUP BY o.date_of_sale
6
7     UNION
8
9     SELECT pu.date_of_purchase AS date_day, 0 AS income, SUM(pl.count * pl.price) AS expense
10    FROM wholesale_base.purchase pu
11    JOIN wholesale_base.purchase_list pl ON pu.purchase_id = pl.purchase_id
12    GROUP BY pu.date_of_purchase
13 )
14 WHERE date_day = CURRENT_DATE - INTERVAL '273 day'
15 GROUP BY date_day;
16
```

Data Output

Messages

Notifications

date_day

profit

date

numeric

1

2023-03-20

13108

- Вычислить общую стоимость каждого вида товара, находящегося на базе.

Query		Query History
1	SELECT p.name AS product_name, SUM (pl.price * pl.count) AS total_cost	
2	FROM wholesale_base.product p	
3	JOIN wholesale_base.purchase_list pl ON p.product_id = pl.product_id	
4	GROUP BY p.product_id, p.name;	

Data Output		Messages	Notifications
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>			
	product_name character varying (50)		total_cost bigint
1	Product CC		33539
2	Product D		21390
3	Product G		33651
4	Product J		22610
5	Product I		24456
6	Product O		245216
7	Product F		41841
8	Product Z		8730
9	Product L		101634

- В какой день было вывезено минимальное количество товара?

Query		Query History
1	SELECT slt.date_of_sale, slt.count_of_product_in_orders FROM (
2	SELECT date_of_sale, count_of_product_in_orders,	
3	RANK () OVER (ORDER BY count_of_product_in_orders) AS rnk	
4	FROM (
5	SELECT o.date_of_sale, SUM (ol.count) AS count_of_product_in_orders	
6	FROM wholesale_base.order o	
7	JOIN wholesale_base.order_list ol ON o.order_id = ol.order_id	
8	GROUP BY o.date_of_sale	
9)	
10	ORDER BY count_of_product_in_orders	
11) slt	
12	WHERE rnk = 1;	

Data Output		Messages	Notifications
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>			
	date_of_sale date		count_of_product_in_orders bigint
1	2023-07-30		14

- Сколько различных видов товара имеется на базе?

Query

Query History

1

SELECT COUNT(*) FROM wholesale_base.product;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	count	
	bigint	🔒
1		30

Представления

- Количество заказов фирм-покупателей за прошедший год

Query

Query History

1

CREATE VIEW number_of_orders_from_customers_for_past_year AS

2

SELECT c.customer_id, c.customer_company_name AS company, COUNT(o.order_id) AS number_of_orc

3

FROM wholesale_base.customer c

4

JOIN wholesale_base.order o ON c.customer_id = o.customer_id

5

AND o.date_of_sale >= CURRENT_DATE - INTERVAL '1 year'

6

GROUP BY c.customer_id, c.customer_company_name;

Query

Query History

1

SELECT

*

FROM

number_of_orders_from_customers_for_past_year;

Data Output

Messages

Notifications

	customer_id integer	company character varying (50)	number_of_orders bigint
1	4	ACME Enterprises	2
2	10	SERENITY Ltd.	2
3	9	HARMONY Corporation	2
4	7	FORTUNE Ltd.	2
5	15	DREAM Corporation	2
6	6	VICTORY Corporation	2
7	12	PEACE Corporation	2
8	14	HOPE Enterprises	1
9	3	TRIUMPH Ltd.	2
10	13	UNITY Ltd.	1

- Доход базы за конкретный период

```

Query    Query History
1  CREATE VIEW company_income_for_year AS
2  SELECT company, income - expenses AS profit FROM (
3      SELECT c.customer_company_name AS company, SUM(ol.price * ol.count) AS income, 0 AS expenses
4      FROM wholesale_base.customer c
5      JOIN wholesale_base.order o ON c.customer_id = o.customer_id
6      AND o.date_of_sale BETWEEN CURRENT_DATE - INTERVAL '3 year' AND CURRENT_DATE
7      JOIN wholesale_base.order_list ol ON o.order_id = ol.order_id
8      GROUP BY c.customer_company_name
9
10     UNION
11
12     SELECT s.company_name AS company, 0 AS income, SUM(pl.price * pl.count) AS expenses
13     FROM wholesale_base.supplier s
14     JOIN wholesale_base.purchase pu ON s.supplier_id = pu.supplier_id
15     AND pu.date_of_purchase BETWEEN CURRENT_DATE - INTERVAL '3 year' AND CURRENT_DATE
16     JOIN wholesale_base.purchase_list pl ON pu.purchase_id = pl.purchase_id
17     GROUP BY s.company_name
18 );

```

Query

Query History

1

```
SELECT SUM(profit) FROM company_income_for_year;
```

Data Output

Messages

Notifications

+
≡

▼

▼

	sum numeric 🔒
1	-1151414

Запросы на модификацию данных

- Insert: добавить покупку товара по айди продукта и его дате производства

```
Query  Query History
1  INSERT INTO wholesale_base.order_list(price, count, order_id, purchase_list_id)
2  VALUES
3  (300, 32, 31, (SELECT purchase_list_id FROM wholesale_base.purchase_list
4                  WHERE product_id = 11
5                  AND manufacture_date >= '2018-01-01'
6                  AND leftovers >= 32
7                  LIMIT 1)
8  );
```

Data Output Messages Notifications

```
INSERT 0 1
```

Query returned successfully in 38 msec.

Query

Query History

1

SELECT * FROM wholesale_base.order_list;

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

📊

⬇️

📈

	order_list_id [PK] integer	price integer	count integer	order_id integer	purchase_list_id integer
52	137	127	33	28	68
53	138	534	62	28	69
54	139	689	62	29	70
55	140	432	40	29	71
56	141	311	60	29	72
57	142	211	34	29	73
58	143	377	34	30	82
59	144	499	32	30	83
60	145	187	22	30	84
61	146	201	71	30	85
62	147	233	91	30	86
63	148	333	21	30	87

Query Query History

```
1 SELECT * FROM wholesale_base.order_list;
```

Data Output Messages Notifications

	order_list_id [PK] integer	price integer	count integer	order_id integer	purchase_list_id integer
54	139	689	62	29	70
55	140	432	40	29	71
56	141	311	60	29	72
57	142	211	34	29	73
58	143	377	34	30	82
59	144	499	32	30	83
60	145	187	22	30	84
61	146	201	71	30	85
62	147	233	91	30	86
63	148	333	21	30	87
64	150	300	32	31	37

- Update: обновить состояние продукта по названию продукта и

Query Query History

```
1 UPDATE wholesale_base.purchase_list
2 SET leftovers = leftovers - 20
3 WHERE leftovers >= 20
4 AND product_id = (SELECT product_id FROM wholesale_base.product
5                   WHERE name='Product M' LIMIT 1)
6 AND manufacture_date = '2018-06-10'
```

Query

Query History

```
1 SELECT * FROM wholesale_base.purchase_list
2 WHERE leftovers >= 20
3 AND product_id = (SELECT product_id FROM wholesale_base.product
4                   WHERE name='Product M' LIMIT 1)
5 AND manufacture_date = '2018-06-10'|
```

Data Output

Messages

Notifications

<

Query

Query History

1

SELECT * FROM wholesale_base.purchase_list

2

WHERE leftovers >= 20

3

AND product_id = (SELECT product_id FROM wholesale_base.product

4

WHERE name='Product M' LIMIT 1)

5

AND manufacture_date = '2018-06-10'

Data Output

Messages

Notifications

<

- Delete: удалить покупки товаров, которые произошли раньше какого-то срока, а также у которых не осталось остатков на складе

Query Query History

```
1 DELETE FROM wholesale_base.order_list
2 WHERE order_id IN (SELECT order_id
3                     FROM wholesale_base.order
4                     WHERE date_of_sale <= '2024-01-01')
5 AND purchase_list_id IN (SELECT purchase_list_id
6                           FROM wholesale_base.purchase_list
7                           WHERE leftovers = 0
8                           );
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 50 msec.

Query Query History

```
1 SELECT o.date_of_sale, ol.* FROM wholesale_base.order_list ol
2 JOIN wholesale_base.order o ON ol.order_id = o.order_id
3 WHERE o.date_of_sale <= '2024-01-01'
4 AND purchase_list_id IN (SELECT purchase_list_id
5                           FROM wholesale_base.purchase_list
6                           WHERE leftovers = 0
7                           );|
```

Data Output Messages Notifications

	date_of_sale date	order_list_id integer	price integer	count integer	order_id integer	purchase_list_id integer
1	2023-12-20	103	55	28	13	29

QueryQuery History

```

1 SELECT o.date_of_sale, ol.* FROM wholesale_base.order_list ol
2 JOIN wholesale_base.order o ON ol.order_id = o.order_id
3 WHERE o.date_of_sale <= '2024-01-01'
4 AND purchase_list_id IN (SELECT purchase_list_id
5                           FROM wholesale_base.purchase_list
6                           WHERE leftovers = 0
7                           );

```

Data OutputMessagesNotifications

date_of_sale	order_list_id	price	count	order_id	purchase_list_id
date	integer	integer	integer	integer	integer

Query History

QueryQuery History

Show queries generated internally by pgAdmin? ☒

RemoveRemove All

Today - 11.12.2023

▶ SELECT o.date_of_sale, ol.* FROM whol...
03:21:01

▶ DELETE FROM wholesale_base.order_list...
03:20:25

▶ SELECT o.date_of_sale, ol.* FROM whol...
03:17:12

▶ SELECT o.date_of_sale, ol.* FROM whol...
03:16:33

▶ SELECT o.date_of_sale, ol.* FROM whol...
03:16:29

▶ SELECT o.date_of_sale, ol.* FROM whol...
03:10:38

11.12.202328 msec

03:21:01

DateRows affectedDuration

CopyCopy to Query Editor

```

SELECT o.date_of_sale, ol.* FROM wholesal
JOIN wholesale_base.order o ON ol.order_id
WHERE o.date_of_sale <= '2024-01-01'
AND purchase_list_id IN (SELECT purchase_
                           FROM wholesale_b
                           WHERE leftovers
                           );

```

Messages

Successfully run. Total query runtime: 28 msec. 0 rows affected.

Data OutputMessagesNotifications

date_of_sale	order_list_id	price	count	order_id	purchase_list_id
date	integer	integer	integer	integer	integer

Индексы

- Обычный индекс

QueryQuery History

```

1 CREATE INDEX idx_purchase_list ON wholesale_base.purchase_list
2 (expiration_date);

```

Рисунок 0.1. создание

Successfully run. Total query runtime: 121 msec.
1 rows affected.

Рисунок 0.2. без индекса

Data Output	Messages	Explain	×	Notifications
Successfully run. Total query runtime: 110 msec. 1 rows affected.				

Рисунок 0.3. с индексом

- Составной индекс индекс

Query	Query History
1	CREATE INDEX idx_purchase_list ON wholesale_base.purchase_list
2	(price, count , expiration_date, leftovers);

Рисунок 0.4. создание

Query	Query History	
1	SELECT pl.price,pl.count,pl.expiration_date,pl.leftovers, pr.name,pr.packing_type	
2	FROM wholesale_base.purchase_list pl	
3	JOIN wholesale_base.product pr ON pl.product_id = pr.product_id	
4	WHERE pl.price BETWEEN 100 AND 400	
5	AND pl.count BETWEEN 35 AND 50	
6	AND pl.expiration_date BETWEEN '2019-01-01' AND '2020-01-01'	
7	AND pl.leftovers BETWEEN 25 AND 50;	
Data Output	Messages	Explain × Notifications
Successfully run. Total query runtime: 85 msec. 1 rows affected.		

Рисунок 0.5. без индекса

```
Query Query History
1 SELECT pl.price,pl.count,pl.expiration_date,pl.leftovers, pr.name,pr.packing_type
2 FROM wholesale_base.purchase_list pl
3 JOIN wholesale_base.product pr ON pl.product_id = pr.product_id
4 WHERE pl.price BETWEEN 100 AND 400
5 AND pl.count BETWEEN 35 AND 50
6 AND pl.expiration_date BETWEEN '2019-01-01' AND '2020-01-01'
7 AND pl.leftovers BETWEEN 25 AND 50;
```

Data Output Messages Explain X Notifications

Successfully run. Total query runtime: 76 msec.
2 rows affected.

Рисунок 0.6. с индексом

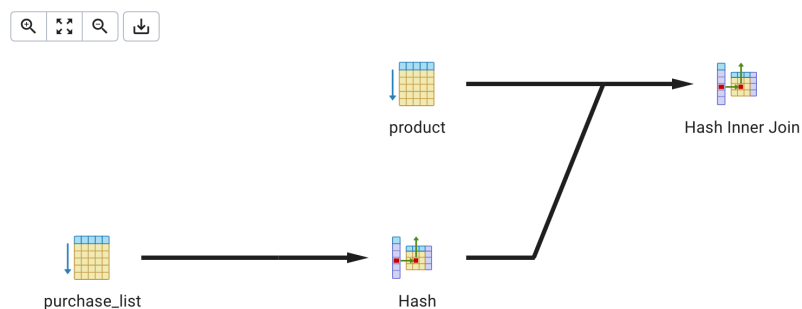


Рисунок 0.7. графическое представление

Вывод

В данной лабораторной работе были задействованы различные sql запросы: агрегации, присоединения, группировки, оконные и другие. Были проведены dml запросы: обычные и с подзапросами. Также в ходе выполнения лабораторной работы были изучены графическое представление запросов через EXPLAIN, а также уделено внимание производительности через создание индексов.