

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:

Дущенко Д.А.

Группа: К3240

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.

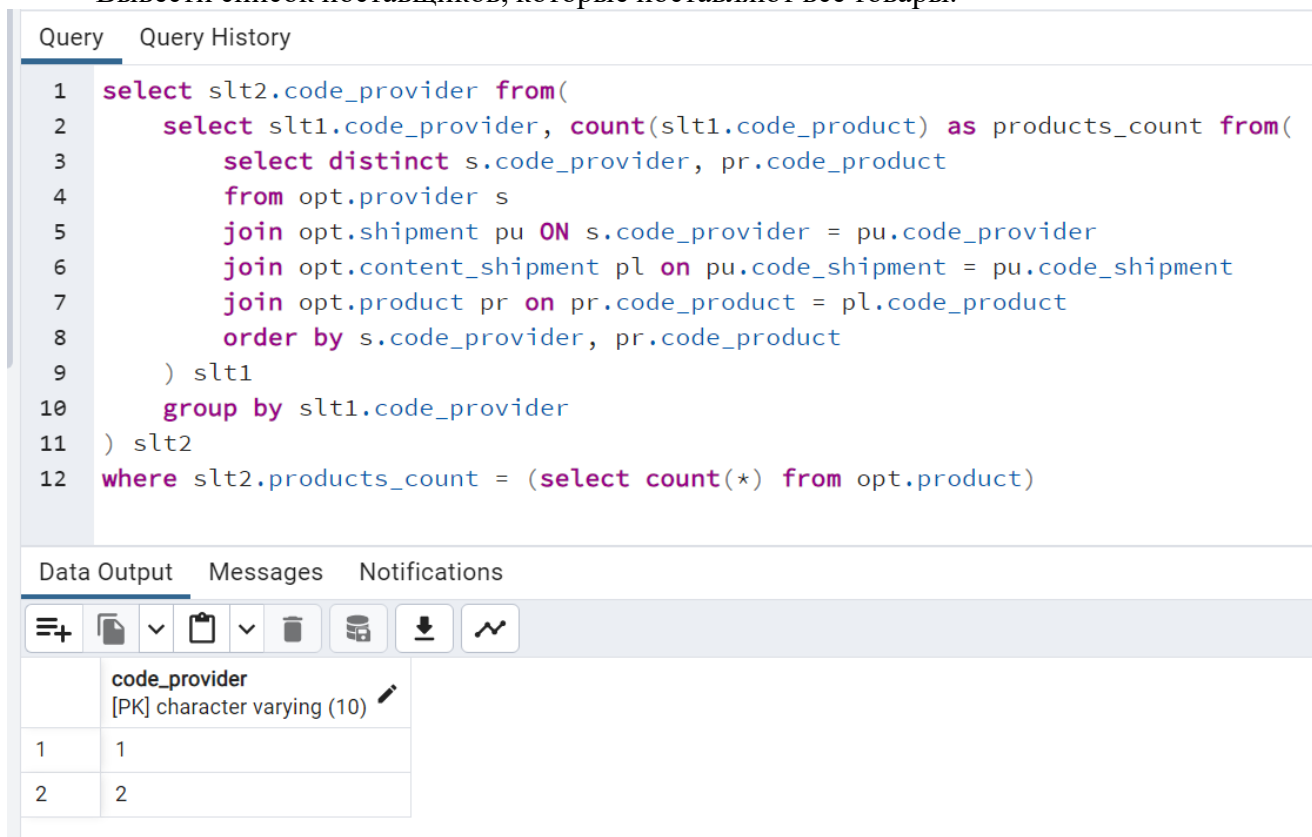
Практическое задание

- 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.



```
1 select slt2.code_provider from(
2     select slt1.code_provider, count(slt1.code_product) as products_count from(
3         select distinct s.code_provider, pr.code_product
4         from opt.provider s
5         join opt.shipment pu ON s.code_provider = pu.code_provider
6         join opt.content_shipment pl on pu.code_shipment = pu.code_shipment
7         join opt.product pr on pr.code_product = pl.code_product
8         order by s.code_provider, pr.code_product
9     ) slt1
10    group by slt1.code_provider
11 ) slt2
12 where slt2.products_count = (select count(*) from opt.product)
```

Data Output Messages Notifications

	code_provider [PK] character varying (10)
1	1
2	2

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

Query

Query History

```
1 select slt.code_provider, slt.code_product, slt.product_price
2 from(
3     select s.code_provider, pr.code_product, pl.product_price,
4     rank() over (partition by pr.code_product order by pl.product_price) as rnk
5     from opt.provider s
6     join opt.shipment pu on s.code_provider = pu.code_provider
7     join opt.content_shipment pl on pu.code_shipment = pl.code_shipment
8     join opt.product pr on pr.code_product = pl.code_product
9     group by s.code_provider, pr.code_product, pl.product_price
10    order by pr.code_product, pl.product_price, s.code_provider
11 ) slt
12 where slt.rnk = 1
13 group by slt.code_provider, slt.code_product, slt.product_price, slt.rnk
14 order by slt.code_product, slt.product_price
```

Data Output

Messages

Notifications

	code_provider character varying (10)	code_product character varying (10)	product_price integer
1	1	1	10
2	2	2	30

- 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_order as date_day, sum(ol.product_amount * ol.product_price) as income, 0 as expenses
3     from opt.order o
4     join opt.order_content ol on o.code_order = ol.code_order
5     group by o.date_order
6
7     union
8
9     select pu.date_shipment as date_day, 0 as income, sum(pl.product_amount * pl.product_price) as expenses
10    from opt.shipment pu
11    join opt.content_shipment pl on pu.code_shipment = pl.code_shipment
12    group by pu.date_shipment
13 )
14 where date_day = current_date - 1
15 group by date_day;
```

Data Output

Messages

Notifications

	date_day date	profit numeric	
1	2024-03-05	50000	

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

Query

Query History

1

select p.name as product_name, sum(pl.product_price * pl.product_amount) as total_cost

2

from opt.product p

3

join opt.content_shipment pl on p.code_product = pl.code_product

4

group by p.code_product, p.name

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	product_name character varying (50) 🔒	total_cost bigint 🔒
1	Молоко 2% 0.8	11000
2	Молоко 3% 0.5	3000

- В какой день было вывезено минимальное количество товара?

Query

Query History

```
1 select slt.date_order, slt.count_of_product_in_order from(
2     select date_order, count_of_product_in_order,
3     rank() over (order by count_of_product_in_order) as rnk
4     from (
5         select o.date_order, sum(ol.product_amount) as count_of_product_in_order
6         from opt. order o
7         join opt.order_content ol on o.code_order = ol.code_order
8         group by o.date_order
9     )
10    order by count_of_product_in_order
11 ) slt
12 where rnk = 1
```

Data Output

Messages

Notifications

date_order

count_of_product_in_order

date

bigint

1

2023-08-13

100

- Сколько различных видов товара имеется на базе?

Query

Query History

1

```
select count(*) from opt.product
```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	count bigint	🔒
1	2	

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.code_buyer, c.name as company, count(o.code_order) as number_of_orders_from_customers_for_past
3 from opt.buyer c
4 join opt.order o on c.code_buyer = o.code_buyer
5 and o.date_order >= current_date - interval '1 year'
6 group by c.code_buyer, c.name

```

Query

Query History










1

select * from number_of_orders_from_customers_for_past_year;

Data Output

Messages

Notifications



	<div>code_buyer</div> <div>character varying (10)</div> <div></div>	<div>company</div> <div>character varying (50)</div> <div></div>	<div>number_of_orders_from_customers_for_past_year</div> <div>bigint</div> <div></div>
1	2	Перекресток	1
2	1	Пятерочка	2

- Доход базы за конкретный период

Query	Query History
1	<code>create view company_income_for_year as</code>
2	<code>select company, income - expenses as profit from (</code>
3	<code>select c.name as company, sum(ol.product_price * ol.product_amount) as income, 0 as expenses</code>
4	<code>from opt.buyer c</code>
5	<code>join opt.order o on c.code_buyer = o.code_buyer</code>
6	<code>and o.date_order between current_date - interval '3 year' and current_date</code>
7	<code>join opt.order_content ol on o.code_order = ol.code_order</code>
8	<code>group by c.name</code>
9	
10	<code>union</code>
11	
12	<code>select s.name as company, 0 as income, sum(pl.product_price * pl.product_amount) as expenses</code>
13	<code>from opt.provider s</code>
14	<code>join opt.shipment pu on s.code_provider = pu.code_provider</code>
15	<code>and pu.date_shipment between current_date - interval '3 year' and current_date</code>
16	<code>join opt.content_shipment pl on pu.code_shipment = pl.code_shipment</code>
17	<code>group by s.name)</code>

Query	Query History
1	<code>select sum(profit) from company_income_for_year;</code>

Data Output	Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	sum	numeric 🔒
1	50000	

Запросы на модификацию данных

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

Query	Query History
1	<code>insert into opt.order_content(code_order_content, code_product, product_price, product_amount, code_ord</code>
2	<code>values</code>
3	<code>(4, 1, 100, 10, 3, (select code_content_shipment from opt.content_shipment</code>
4	<code>where code_product = '1'</code>
5	<code>and expiration_date >= '2024-01-01'</code>
6	<code>and rest_shipment >= 10</code>
7	<code>limit 1)</code>
8	<code>);</code>

- Update: обновить состояние продукта по названию продукта и

Query	Query History
1	<code>update opt.content_shipment</code>
2	<code>set rest_shipment = rest_shipment - 10</code>
3	<code>where rest_shipment >= 10</code>
4	<code>and code_product = (select code_product from opt.product</code>
5	<code>where name='Молоко 2% 0.8' limit 1)</code>
6	<code>and expiration_date = '2025-04-01'</code>

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

Query	Query History
1	<code>delete from opt.order_content</code>
2	<code>where code_order in (select code_order</code>
3	<code>from opt.order</code>
4	<code>where date_order <= '2023-12-12')</code>
5	<code>and code_content_shipment in (select code_content_shipment</code>
6	<code>from opt.content_shipment</code>
7	<code>where rest_shipment = 0</code>
8	<code>);</code>

Query History

Query	Query History
Show queries generated internally by pgAdmin? <input checked="" type="checkbox"/> <div>Remove Remove All</div>	
Today - 10.03.2024	
<div> <div> <div></div> <div>select pl.product_price, pl.product_amount, pl...</div> <div>20:20:22</div> </div> <div> <div>▶</div> <div>select pl.product_price, pl.product_amount, pl.e...</div> <div>20:18:12</div> </div> <div> <div>▶</div> <div>select sum(profit) from company_income_for_year;</div> <div>20:09:05</div> </div> <div> <div>▶</div> <div>select * from number_of_orders_from_customers_fo...</div> <div>20:08:26</div> </div> <div> <div>▶</div> <div>select count(*) from opt.product</div> <div>20:07:48</div> </div> <div> <div>▶</div> <div>select slt.date_order, slt.count_of_product_in_o...</div> <div>20:07:13</div> </div> <div> <div>▶</div> <div>select p.name as prduct_name, sum(pl.product_pr...</div> <div>20:07:03</div> </div> <div> <div>▶</div> <div>select p.name as prduct_name, sum(pl.product_pr...</div> <div>20:06:50</div> </div> <div> <div>▶</div> <div>select date_day, sum(income-expenses) as profit ...</div> <div>20:06:15</div> </div> </div>	<div> <div>10.03.2024 20:20:22</div> <div>1</div> <div>72 msec</div> </div> <div> <div>Date</div> <div>Rows affected</div> <div>Duration</div> </div> <div> <div>Copy</div> <div>Copy to Query Editor</div> </div> <div> <pre>select pl.product_price, pl.product_amount, pl.expir from opt.content_shipment pl join opt.product pr on pl.code_product = pr.code_pro where pl.product_price between 10 and 500 and pl.product_amount between 2 and 100 and pl.expiration_date between '2020-01-01' and '202 and pl.rest_shipment between 10 and 100</pre> </div> <div> <div>Messages</div> <div>Successfully run. Total query runtime: 72 msec. 1 rows affected.</div> </div>

Индексы

- Обычный индекс

Query	Query History
1	<code>create index idx_content_shipment on opt.content_shipment</code>
2	<code>("expiration_date");</code>

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	<code>create index idx_content_shipment2 on opt.content_shipment</code>
2	<code>(product_price, product_amount, expiration_date, rest_shipment);</code>

Рисунок 0.4. создание

Query	Query History
1	<code>select pl.product_price, pl.product_amount, pl.expiration_date, pl.rest_shipment, pr.name</code>
2	<code>from opt.content_shipment pl</code>
3	<code>join opt.product pr on pl.code_product = pr.code_product</code>
4	<code>where pl.product_price between 10 and 500</code>
5	<code>and pl.product_amount between 2 and 100</code>
6	<code>and pl.expiration_date between '2020-01-01' and '2025-01-01'</code>
7	<code>and pl.rest_shipment between 10 and 100</code>

Data Output	Messages	Explain	×	Notifications
Successfully run. Total query runtime: 85 msec. 1 rows affected.				

Рисунок 0.5. без индекса

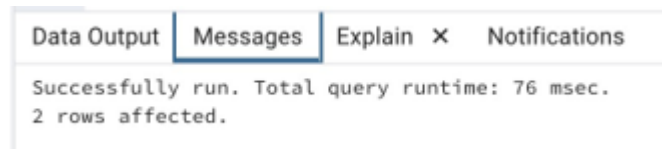


Рисунок 0.6. с индексом

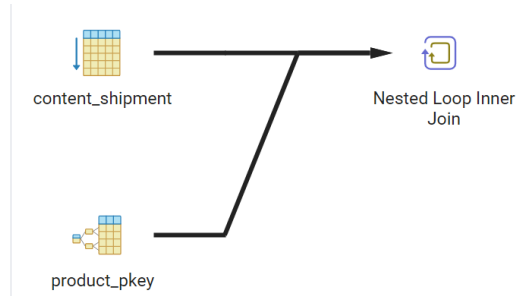


Рисунок 0.7. графическое представление

Вывод

В данной лабораторной работе были задействованы различные sql запросы: агрегации, соединения, 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.