**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 K32392

Тишалович Леонид Михайлович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(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.

**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 посмотреть 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.

**Индивидуальное практическое задание:**

База данных “Оптовая база”

Запросы на выборку:

1. Вывести список поставщиков, которые поставляют все товары.
2. Определить поставщика, который поставляет каждый из товаров по самой низкой цене.
3. Вывести названия товаров, цены на которые у всех поставщиков одинаковы.
4. Чему равен общий суточный доход оптового склада за прошедший день?
5. Вычислить общую стоимость каждого вида товара, находящегося на базе.
6. В какой день было вывезено минимальное количество товара?
7. Сколько различных видов товара имеется на базе?

Представления:

1. количество заказов фирм-покупателей за прошедший год;
2. доход базы за конкретный период.

**Выполнение**

**Запросы на выборку:**

1. Вывести список поставщиков, которые поставляют все товары.

SELECT p.name\_of\_company

FROM provider p

JOIN shipment s ON p.provider\_id = s.provider\_id

JOIN shipment\_content sc ON s.shipment\_id = sc.shipment\_id

JOIN product pr ON sc.product\_id = pr.product\_id

GROUP BY p.provider\_id, p.name\_of\_company

HAVING COUNT(DISTINCT pr.product\_id) = (SELECT COUNT(DISTINCT pr1.product\_id) FROM product pr1);

Рис. 1 - SELECT №1

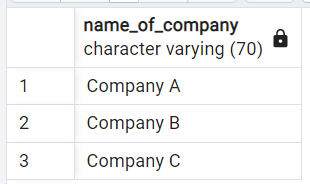


Рис. 2 - Результат SELECT №1

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

SELECT

p.name\_of\_company,

pr.product\_name,

MIN(sc.price) AS lowest\_price

FROM

provider p

JOIN shipment s ON p.provider\_id = s.provider\_id

JOIN shipment\_content sc ON s.shipment\_id = sc.shipment\_id

JOIN product pr ON sc.product\_id = pr.product\_id

WHERE

sc.price = (

SELECT

MIN(sc2.price)

FROM

shipment\_content sc2

JOIN shipment s2 ON sc2.shipment\_id = s2.shipment\_id

WHERE

s2.provider\_id = p.provider\_id

AND sc2.product\_id = sc.product\_id

)

GROUP BY

p.name\_of\_company, pr.product\_name;

Рис. 3 - SELECT №2

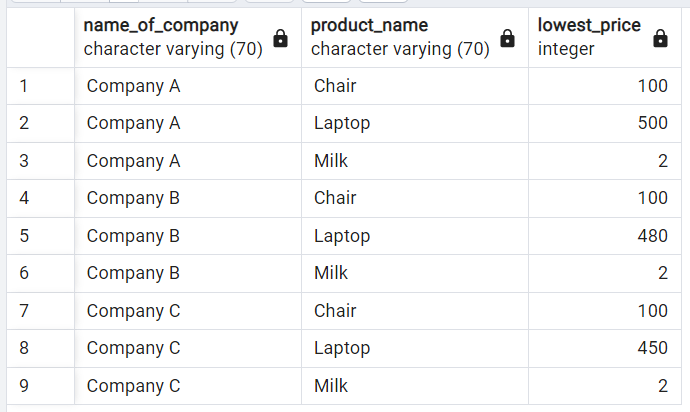


Рис. 4 - Результат SELECT №2

3. Вывести названия товаров, цены на которые у всех поставщиков одинаковы.

SELECT

pr.product\_name,

sc.price

FROM

product pr

JOIN shipment\_content sc ON pr.product\_id = sc.product\_id

JOIN shipment s ON sc.shipment\_id = s.shipment\_id

JOIN provider p ON s.provider\_id = p.provider\_id

WHERE

sc.price = (

SELECT

MIN(sc2.price)

FROM

shipment\_content sc2

JOIN shipment s2 ON sc2.shipment\_id = s2.shipment\_id

WHERE

sc2.product\_id = sc.product\_id

)

GROUP BY

pr.product\_name, sc.price

HAVING

COUNT(DISTINCT p.name\_of\_company) = (

SELECT

COUNT(\*)

FROM

provider

);

Рис. 5 - SELECT №3

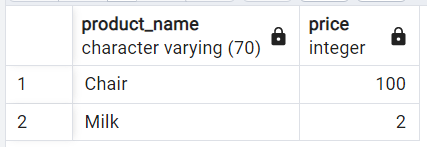


Рис. 6 - Результат SELECT №3

4. Чему равен общий суточный доход оптового склада за прошедший день?

SELECT

SUM(pc.amount\_of\_goods \* sc.price) AS daily\_income

FROM

purchase\_content pc

JOIN purchase\_invoice pi ON pc.purchase\_id = pi.purchase\_id

JOIN purchase p ON pc.purchase\_id = p.purchase\_id

JOIN shipment\_content sc ON pc.product\_id = sc.product\_id

WHERE

pi.invoice\_date = CURRENT\_DATE

AND p.status = 'completed';

Рис. 7 - SELECT №4

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

SELECT

p.product\_name,

SUM(sc.remains \* sc.price) AS total\_cost

FROM

product p

JOIN shipment\_content sc ON p.product\_id = sc.product\_id

WHERE

sc.remains > 0

GROUP BY

p.product\_name;

Рис. 8 - SELECT №5

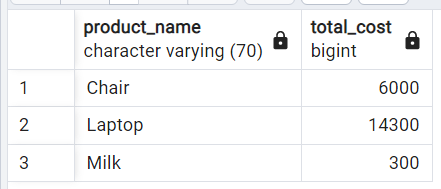


Рис. 9 - Результат SELECT №5

6. В какой день было вывезено минимальное количество товара?

SELECT

sc.expiration\_date,

SUM(sc.amount\_of\_goods) AS total\_goods

FROM

shipment\_content sc

JOIN shipment s ON sc.shipment\_id = s.shipment\_id

WHERE

s.invoice\_fulfillment\_date = (

SELECT MIN(invoice\_fulfillment\_date)

FROM shipment

)

GROUP BY

sc.expiration\_date

ORDER BY

total\_goods ASC

LIMIT 1;

Рис. 10 - SELECT №6

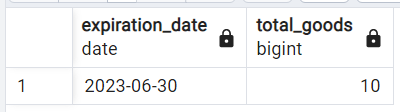


Рис. 11 - Результат SELECT №6

7. Сколько различных видов товара имеется на базе?

SELECT

COUNT(\*) AS num\_products

FROM

product;

Рис. 12 - SELECT №7

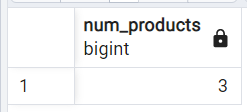


Рис. 13 - Результат SELECT №7

**Представления:**

1.Количество заказов фирм-покупателей за прошедший год;

CREATE VIEW num\_orders\_last\_year AS

SELECT

c.organization\_name,

COUNT(\*) AS num\_orders

FROM

purchase p

JOIN client c ON p.client\_id = c.client\_id

WHERE

p.invoice\_fulfillment\_date >= DATE\_TRUNC('year', CURRENT\_DATE - INTERVAL '1 year')

AND p.invoice\_fulfillment\_date < DATE\_TRUNC('year', CURRENT\_DATE)

GROUP BY

c.organization\_name;

Рис. 14 - VIEW №1

2.Доход базы за конкретный период.

CREATE VIEW base\_income AS

SELECT

SUM(sc.remains \* sc.price) AS income

FROM

shipment\_content sc

JOIN shipment s ON sc.shipment\_id = s.shipment\_id

WHERE

s.invoice\_fulfillment\_date >= '2023-01-01'

AND s.invoice\_fulfillment\_date <= '2023-04-30';

Рис. 15 - VIEW №1

**INSERT:**

INSERT INTO purchase\_invoice (purchase\_id, invoice\_number, invoice\_date, payment\_date)

SELECT

p.purchase\_id,

100,

'2023-05-01',

'2023-05-10'

FROM

purchase p

JOIN client c ON p.client\_id = c.client\_id

WHERE

c.organization\_name IN (

SELECT organization\_name

FROM purchase p

JOIN client c ON p.client\_id = c.client\_id

WHERE

p.invoice\_fulfillment\_date >= DATE\_TRUNC('year', CURRENT\_DATE - INTERVAL '1 year')

AND p.invoice\_fulfillment\_date < DATE\_TRUNC('year', CURRENT\_DATE)

);

Рис. 16 - INSERT

**UPDATE:**

UPDATE shipment\_content

SET

remains = 10

WHERE

product\_id = (

SELECT product\_id

FROM product

WHERE product\_name = 'Kettle'

);

Рис. 17 - UPDATE

**DELETE:**

DELETE FROM shipment

WHERE

manager\_id = (

SELECT manager\_id

FROM manager

WHERE contact\_information IS NULL

);

Рис. 18 - DELETE

**Простой индекс:**

SELECT p.name\_of\_company

FROM provider p

JOIN shipment s ON p.provider\_id = s.provider\_id

JOIN shipment\_content sc ON s.shipment\_id = sc.shipment\_id

JOIN product pr ON sc.product\_id = pr.product\_id

GROUP BY p.provider\_id, p.name\_of\_company

HAVING COUNT(DISTINCT pr.product\_id) = (SELECT COUNT(DISTINCT pr1.product\_id) FROM product pr1);

CREATE INDEX idx\_provider\_name\_of\_company

ON provider (name\_of\_company);

Рис. 19 - простой INDEX

Проведя серию замеров из 20 измерений, получилось сократить среднее время выполнения с 87мс до 77мс.

**Составной индекс:**

SELECT

p.name\_of\_company,

pr.product\_name,

MIN(sc.price) AS lowest\_price

FROM

provider p

JOIN shipment s ON p.provider\_id = s.provider\_id

JOIN shipment\_content sc ON s.shipment\_id = sc.shipment\_id

JOIN product pr ON sc.product\_id = pr.product\_id

WHERE

sc.price = (

SELECT

MIN(sc2.price)

FROM

shipment\_content sc2

JOIN shipment s2 ON sc2.shipment\_id = s2.shipment\_id

WHERE

s2.provider\_id = p.provider\_id

AND sc2.product\_id = sc.product\_id

)

GROUP BY

p.name\_of\_company, pr.product\_name;

CREATE INDEX idx\_shipment\_content\_shipment\_id\_product\_id

ON shipment\_content (shipment\_id, product\_id);

Рис. 20 - составной INDEX

Проведя серию замеров из 30 измерений, получилось сократить среднее время выполнения с 954мс до 872мс.

**Выводы:**

В процессе выполнения лабораторной работы получилось ознакомиться с составлением INSERT, UPDATE, DELETE запросов. Также удалось ознакомиться с графическим представлением запросов. Были составлены простые и составные индексы, что позволило наглядно увидеть уменьшение кол-ва этапов при выполнении запроса, а также уменьшение времени выполнения.