

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по Лабораторной Работе №4

по дисциплине «**БазыДанных**»

Вариант 9. Оптовая база

Автор: Чебан Илья Валерьевич

Факультет: ФИКТ

Группа: K3241

Преподаватель: Говорова Марина Михайловна

**ИТМО**

Санкт-Петербург

2023

**1. Цель работы:** Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**2. Практическое задание:**

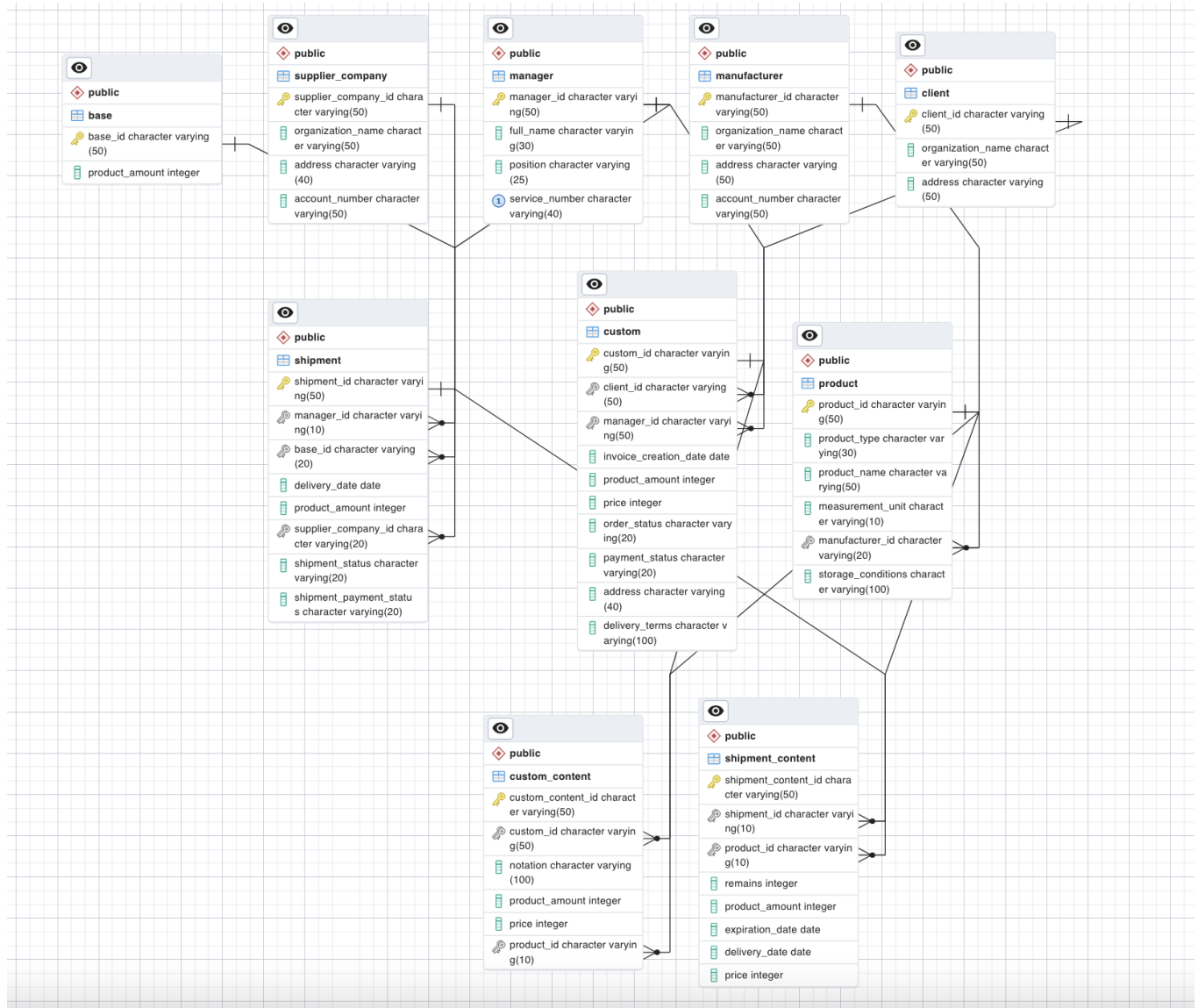
1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).

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.

**3. Схема базы данных:**



#### 4. Выполнение:

##### Создать запросы:

- 1) Вывести список поставщиков, которые поставляют все товары:
- 2) Определить поставщика, который поставляет каждый из товаров по самой низкой цене.

```

SELECT
    sc.organization_name,
    pr.product_name,
    MIN(sct.price) AS lowest_price
FROM
    supplier_company sc
    JOIN shipment s ON sc.supplier_company_id = s.supplier_company_id
    JOIN shipment_content sct ON s.shipment_id = sct.shipment_id
    JOIN product pr ON sct.product_id = pr.product_id
WHERE
    sct.price = (
        SELECT
            MIN(sct2.price)
        FROM
  
```

```
shipment_content sct2
JOIN shipment s2 ON sct2.shipment_id = s2.shipment_id
WHERE
s2.supplier_company_id = sc.supplier_company_id
AND sct2.product_id = sct.product_id
)
GROUP BY
sc.organization_name, pr.product_name;
```

|   | <b>organization_name</b><br>character varying (50) 🔒 | <b>product_name</b><br>character varying (50) 🔒 | <b>lowest_price</b><br>integer 🔒 |
|---|--|---|----------------------------------|
| 1 | A1   | Product A                                       | 250                              |
| 2 | A2   | Product B                                       | 375                              |
| 3 | A3   | Product C                                       | 500                              |

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

```

SELECT pr.product_name, sct.price
FROM product pr
JOIN shipment_content sct ON pr.product_id = sct.product_id
JOIN shipment s ON s.shipment_id = sct.shipment_id
JOIN manager m ON m.manager_id = s.manager_id
WHERE
sct.price = (
  SELECT MIN(sct2.price) FROM shipment_content sct2
  JOIN shipment sc2 ON s.shipment_id = sct2.shipment_id
  WHERE sct2.product_id = sct.product_id
)
GROUP BY pr.product_name, sct.price;

```

| Data Output |   |                           | Messages | Notifications |
|-------------|---|---------------------------|----------|---------------|
|             | <b>product_name</b><br>character varying (50) 🔒 | <b>price</b><br>integer 🔒 |          |               |
| 1           | Product A                                       | 250                       |          |               |
| 2           | Product B                                       | 375                       |          |               |
| 3           | Product C                                       | 500                       |          |               |

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

```


SELECT SUM(csc.product_amount * sct.price) AS daily_amount
FROM custom_content csc

```



```

JOIN custom cs ON cs.custom_id = csc.custom_id
JOIN shipment_content sct ON sct.product_id = csc.product_id
WHERE
cs.invoice_creation_date = CURRENT_DATE AND cs.order_status = 'completed'

```


|   | <b>daily_amount</b><br>bigint  |
|---|--|
| 1 | [null]   |

- 5) Вычислить общую стоимость каждого вида товара, находящегося на базе.  
 SELECT p.product\_name, SUM(sc.remains \* sc.price) AS total\_amount  
 FROM product p  
 JOIN shipment\_content sc ON p.product\_id = sc.product\_id  
 WHERE sc.remains > 0  
 GROUP BY p.product\_name

|   | <b>product_name</b><br>character varying (50)  | <b>total_amount</b><br>bigint  |
|---|---|---|
| 1 | Product B   | 28125   |
| 2 | Product A   | 12500   |
| 3 | Product C   | 50000   |



- 6) В какой день было вывезено минимальное количество товара?  
пока не осилил
- 7) Сколько различных видов товара имеется на базе?  
`SELECT COUNT(*) FROM product`

|   | count<br>bigint |  |
|---|-----------------|---|
| 1 | 103             |   |



## Создать представление:


- 1) количество заказов фирм-покупателей за прошедший год.

```
CREATE VIEW num_customs_in_last_year AS
SELECT c.organization_name, COUNT(*) AS num_customs
FROM custom cs
JOIN client c ON cs.client_id = c.client_id
WHERE cs.invoice_creation_date >= DATE_TRUNC('year', CURRENT_DATE - INTERVAL '1 year')
AND cs.invoice_creation_date < DATE_TRUNC('year', CURRENT_DATE)
GROUP BY c.organization_name
```

|  |  |  |
|--|--|--|
|  | <b>organization_name</b><br>character varying (50)  | <b>num_customs</b><br>bigint  |
|--|--|--|

2) доход базы за конкретный период.

```
CREATE VIEW period_income AS  
SELECT SUM(sct.remains * sct.price) AS income  
FROM shipment_content sct  
JOIN shipment s ON sct.shipment_id = s.shipment_id  
WHERE s.delivery_date >= '2023-01-01' AND s.delivery_date <= '2023-12-31'
```

|   | income<br>bigint  |
|---|--|
| 1 | 90625  |

## Запросы на модификацию данных:

### 1) Insert (Добавим состав поставки):

```
INSERT INTO custom_content (custom_content_id, custom_id, notation,  
product_amount, price, product_id)
```

```
SELECT
```

```
'content4',
```

```
cs.custom_id,
```

```
'Хранить при комнатной температуре',
```

```
cs.product_amount,
```

```
cs.price,
```

```
'prod1'
```

```
FROM
```

```
custom cs
```

```
JOIN client cl ON cl.client_id = cs.client_id
```

```
WHERE
```

```
cs.custom_id IN (
```

```
SELECT c.custom_id FROM manager mn
```

```
JOIN custom c ON mn.manager_id = c.manager_id
```

```
WHERE mn.manager_id = 'manager1'
```

```
);
```

### До запроса:

|   | custom_content_id<br>[PK] character varying (50) | custom_id<br>character varying (50) | notation<br>character varying (100) | product_amount<br>integer | price<br>integer | product_id<br>character var |
|---|--|-------------------------------------|-------------------------------------|---------------------------|------------------|-----------------------------|
| 1 | content1   | custom1                             | sample notation 1                   | 50                        | 250              | prod1                       |
| 2 | content2   | custom2                             | sample notation 2                   | 75                        | 375              | prod2                       |
| 3 | content3   | custom3                             | sample notation 3                   | 100                       | 500              | prod3                       |

### После запроса:

|   | <div>custom_content_id</div> <div>[PK] character varying (50)</div> | <div>custom_id</div> <div>character varying (50)</div> | <div>notation</div> <div>character varying (100)</div> | <div>product_amount</div> <div>integer</div> | <div>price</div> <div>integer</div> | <div>product</div> <div>character</div> |
|---|---|--|--|--|-------------------------------------|---|
| 1 | content1  | custom1  | sample notation 1                                      | 50   | 250                                 | prod1                                   |
| 2 | content2  | custom2  | sample notation 2                                      | 75   | 375                                 | prod2                                   |
| 3 | content3  | custom3  | sample notation 3                                      | 100  | 500                                 | prod3                                   |
| 4 | 7fb2b254-a87e-46df-a370-13812ef49381                                | [null]   | in porttitor pede justo eu massa donec dapibus duis at | 93   | 41                                  | [null]                                  |
| 5 | custom_content1   | custom1  | Хранить при комнатной температуре                      | 100  | 500                                 | prod1                                   |
| 6 | content4  | custom1  | Хранить при комнатной температуре                      |  |                                     |   |

**2) Update (Изменим количество товаров на базе 1, которая использовалась для хранения поставок)**

```
UPDATE base
SET product_amount = 10
WHERE base_id = (
    SELECT base_id FROM shipment sh
    WHERE sh.base_id = 'base1'
)
```

**До запроса:**

|     |       |     |
|-----|-------|-----|
| 103 | base2 | 150 |
| 104 | base3 | 200 |
| 105 | base1 | 5   |

**После запроса:**

|     |       |     |
|-----|-------|-----|
| 103 | base2 | 150 |
| 104 | base3 | 200 |
| 105 | base1 | 10  |

### 3) Delete (Удалим из состава покупок запись с покупкой доставленной на базу 2):

```
DELETE FROM shipment_content
WHERE shipment_id = (
    SELECT shipment_id FROM shipment sh
    WHERE sh.base_id = 'base2'
)
```

#### До запроса:

Тут я забыл сделать скрин до, но уверяю там была строка с айди shipment2

#### После запроса:

|     |                   |           |       |     |     |            |            |                |
|-----|-------------------|-----------|-------|-----|-----|------------|------------|----------------|
| 101 | shipment_content1 | shipment1 | prod1 | 50  | 100 | 2023-12-31 | 2023-10-15 | 250            |
| 102 | shipment_content3 | shipment3 | prod3 | 100 | 200 | 2023-12-31 | 2023-10-15 | ✓ Query return |

## Создание индексов:

### 1. Первый индекс

| Query | Query History   |
|-------|---|
| 1     | <code>CREATE INDEX idx_product_id ON public.product (product_id)</code> |

Без индекса:

```
1 EXPLAIN ANALYZE
2 SELECT * FROM product WHERE product.product_id = 'prod1'
```

Data Output Messages Notifications



QUERY PLAN

text



|   |  |
|---|--|
| 1 | Seq Scan on product (cost=0.00..5.25 rows=1 width=164) (actual time=0.246..0.618 rows=1 loops... |
| 2 | Filter: ((product_id)::text = 'prod1'::text)   |
| 3 | Rows Removed by Filter: 102  |
| 4 | Planning Time: 9.525 ms  |
| 5 | Execution Time: 0.949 ms   |



С индексом:

Query Query History

```
1 EXPLAIN ANALYZE
2 SELECT * FROM product WHERE product.product_id = 'prod1'|
```

Data Output Messages Notifications



|   |  |   |
|---|--|---|
|   | QUERY PLAN<br>text   | 🔒 |
| 1 | Seq Scan on product (cost=0.00..5.29 rows=1 width=164) (actual time=0.047..0.142 rows=1 loops... |   |
| 2 | Filter: ((product_id)::text = 'prod1'::text)   |   |
| 3 | Rows Removed by Filter: 102  |   |
| 4 | Planning Time: 4.073 ms  |   |
| 5 | Execution Time: 0.770 ms   |   |

## 2. Второй индекс:

```
1 CREATE INDEX idx_shipme ON public.shipment (base_id, manager_id)
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 70 msec.

Без индекса:

Query Query History

```
1 EXPLAIN ANALYZE
2 SELECT * FROM shipment
3 WHERE
4 shipment.manager_id = 'manager1' AND shipment.base_id = 'base2'
```

Data Output Messages Notifications



|   | QUERY PLAN  |  |
|---|---|--|
|   | text  |  |
| 1 | Seq Scan on shipment (cost=0.00..3.50 rows=1 width=243) (actual time=0.767..0.967 rows=0 loops... |  |
| 2 | Filter: (((manager_id)::text = 'manager1'::text) AND ((base_id)::text = 'base2'::text))           |  |
| 3 | Rows Removed by Filter: 104   |  |
| 4 | Planning Time: 3.288 ms   |  |
| 5 | Execution Time: 1.901 ms  |  |

С индексом:

QueryQuery History

1EXPLAIN ANALYZE

2SELECT \* FROM shipment

3WHERE

4shipment.manager\_id = 'manager1' AND shipment.base\_id = 'base2'

Data OutputMessagesNotifications

QUERY PLAN

text

1

Seq Scan on shipment (cost=0.00..3.56 rows=1 width=243) (actual time=0.042..0.044 rows=0 loops...

2

Filter: (((manager\_id)::text = 'manager1'::text) AND ((base\_id)::text = 'base2'::text))

3

Rows Removed by Filter: 104

4

Planning Time: 2.176 ms

5

Execution Time: 0.078 ms

**Вывод:**

В ходе выполнения лабораторной работы были активно использованы различные типы SQL запросов, включая операции группировки, фильтрации, агрегации, сортировки и присоединений. Были проведены как операции чтения, так и операции записи, удаления и обновления данных с использованием подзапросов. Особое внимание было уделено сравнению производительности запросов до и после применения индексов. Исследование показало значительное улучшение скорости выполнения операций чтения после создания индексов.