

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

**Отчет**

по лабораторной работе №4 «ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ.  
ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ»

по дисциплине «Проектирование и реализация баз данных»

Автор: Цатинян А.А.

Факультет: ПИН

Группа: K3239

Преподаватель: Говорова М.М.

**ИТМО**

Санкт-Петербург 2025

## Оглавление

Цель работы.....	3
Практическое задание .....	3
Выполнение .....	3
Вывод .....	Ошибка! Закладка не определена.

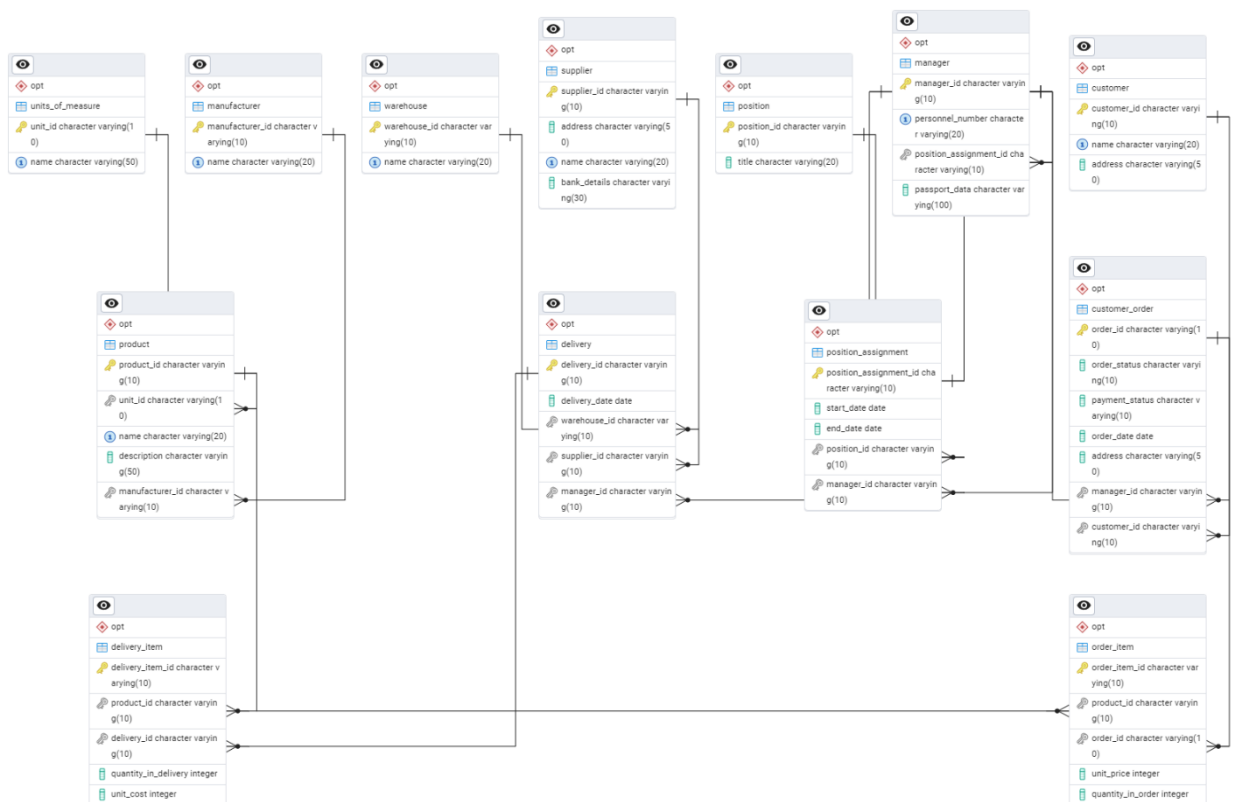
## Цель работы

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 лабораторной работы №2, 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.

## Схема базы данных



## Выполнение

### Вариант 9. БД «Оптовая база»

#### Запросы

Вывести список поставщиков, которые поставляют все товары.

```

SELECT s.supplier_id,
       s.name AS supplier_name
FROM opt.supplier s
      JOIN opt.delivery d ON s.supplier_id = d.supplier_id
      JOIN opt.delivery_item di ON d.delivery_id = di.delivery_id
GROUP BY s.supplier_id, s.name
HAVING COUNT(DISTINCT di.product_id) = (
    SELECT COUNT(*) FROM opt.product
);

```

Data Output Messages Notifications		
<div> <div>SQL</div> <div> <div>+</div> <div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗑️</div> <div>📥</div> <div>📥</div> <div>📈</div> </div> </div> </div>		
	supplier_id [PK] character varying (10)	supplier_name character varying (20)
1	SUP_ALL	Universal Exporters

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

```

SELECT
    p.name AS product_name,
    s.name AS supplier_name,
    di.unit_cost AS minimal_cost
FROM opt.product p
      JOIN opt.delivery_item di ON p.product_id = di.product_id
      JOIN opt.delivery d ON di.delivery_id = d.delivery_id
      JOIN opt.supplier s ON d.supplier_id = s.supplier_id
WHERE di.unit_cost = (
    SELECT MIN(di2.unit_cost)
    FROM opt.delivery_item di2
          JOIN opt.delivery d2 ON di2.delivery_id = d2.delivery_id
    WHERE di2.product_id = p.product_id
);

```

	product_name character varying (20)	supplier_name character varying (20)	minimal_cost integer
1	Galaxy S20	AsiaSupply	250
2	Galaxy S20	Universal Exporters	250
3	OLED TV	AsiaSupply	1100
4	OLED TV	Universal Exporters	1100
5	Rice bag	AsiaSupply	20
6	Rice bag	Universal Exporters	20
7	Juice 1L	AsiaSupply	3
8	Juice 1L	Universal Exporters	3
9	iPhone 13	TechSupplier	700
10	iPhone 13	Universal Exporters	700
11	Milk 2% 1L	GlobalCorp	2
12	Milk 2% 1L	Universal Exporters	2
13	PlayStation 5	WestTrading	499
14	PlayStation 5	Universal Exporters	499
15	Sugar 1kg	AsiaSupply	5
16	Sugar 1kg	Universal Exporters	5

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

```

WITH LastSupplierPrice AS (
    SELECT
        di.product_id,
        d.supplier_id,
        di.unit_cost,
        ROW_NUMBER() OVER(PARTITION BY di.product_id, d.supplier_id
                           ORDER BY d.delivery_date DESC, d.delivery_id DESC) as rn
    FROM opt.delivery_item di
         JOIN opt.delivery d ON di.delivery_id = d.delivery_id
)
SELECT
    p.name AS product_name
FROM LastSupplierPrice lsp
     JOIN opt.product p ON lsp.product_id = p.product_id
WHERE lsp.rn = 1
GROUP BY lsp.product_id, p.name
HAVING MIN(lsp.unit_cost) = MAX(lsp.unit_cost)

```

	product_name character varying (20)
1	Galaxy S20
2	Rice bag
3	Juice 1L
4	iPhone 13
5	Milk 2% 1L
6	PlayStation 5
7	Sugar 1kg

Чему равен общий суточный доход оптового склада за прошедший день?

```

SELECT SUM(oi.quantity_in_order * oi.unit_price) AS daily_revenue
FROM opt.order_item oi
     JOIN opt.customer_order co ON oi.order_id = co.order_id
WHERE co.order_date = CURRENT_DATE - INTERVAL '1 day';

```

Data Output		Messages	Notifications
	daily_revenue bigint		
1	1510		

Вычислить общую стоимость каждого вида товара, находящегося на базе.

```

SELECT p.name AS product_name,
       SUM(di.quantity_in_delivery * di.unit_cost) AS total_cost
FROM opt.delivery_item di
     JOIN opt.delivery d ON di.delivery_id = d.delivery_id
     JOIN opt.product p ON di.product_id = p.product_id
GROUP BY p.name;

```

	product_name character varying (20)	total_cost bigint
1	Sugar 1kg	175
2	Rice bag	400
3	iPhone 13	9100
4	Milk 2% 1L	120
5	PlayStation 5	5988
6	OLED TV	19200
7	Galaxy S20	11000
8	Juice 1L	75

В какой день было вывезено минимальное количество товара?

```
SELECT co.order_date,
       SUM(oi.quantity_in_order) AS total_qty
FROM opt.customer_order co
     JOIN opt.order_item oi ON co.order_id = oi.order_id
GROUP BY co.order_date
ORDER BY total_qty
LIMIT 1;
```

	order_date date	total_qty bigint
1	2023-08-15	2

Сколько различных видов товара имеется на базе?

```
SELECT COUNT(product_id) AS products_in_stock_count
FROM (
    SELECT product_id, SUM(quantity_change) AS current_stock
    FROM (
        SELECT product_id, quantity_in_delivery AS quantity_change FROM
opt.delivery_item
        UNION ALL
        SELECT product_id, -quantity_in_order AS quantity_change FROM opt.order_item
    ) AS movements
    GROUP BY product_id
) AS stock_balance
WHERE current_stock > 0;
```

	products_in_stock_count bigint
1	3

**Создать представления**

Количество заказов фирм-покупателей за прошедший год

```
CREATE OR REPLACE VIEW opt.v_orders_year AS
SELECT c.name AS customer_name,
       COUNT(co.order_id) AS orders_count
```

```
FROM opt.customer_order co
      JOIN opt.customer c ON co.customer_id = c.customer_id
WHERE co.order_date >= (SELECT MAX(order_date) - INTERVAL '1 year' FROM opt.customer_order)
      AND co.order_date <= (SELECT MAX(order_date) FROM opt.customer_order)
GROUP BY c.name;
```

CREATE VIEW

Query returned successfully in 46 msec.

```
SELECT * FROM opt.v_orders_year;
```

Data Output Messages Notifications		
	customer_name character varying (20)	orders_count bigint
1	BestStore	1
2	FoodMart	1
3	HomeDecor	1
4	ITSolutions	1
5	JohnDoe	1
6	MegaCorp	1

Доход базы за конкретный период

```
CREATE OR REPLACE VIEW opt.v_income_period AS
SELECT
      MIN(co.order_date) AS start_period,
      MAX(co.order_date) AS end_period,
      SUM(oi.quantity_in_order * oi.unit_price) AS total_income
FROM opt.customer_order co
      JOIN opt.order_item oi ON co.order_id = oi.order_id
WHERE co.order_date BETWEEN '2023-08-15' AND '2023-08-20';
```

CREATE VIEW

Query returned successfully in 51 msec.

```
SELECT * FROM opt.v_income_period;
```

Data Output Messages Notifications			
	start_period date	end_period date	total_income bigint
1	2023-08-15	2023-08-20	6754

Запросы на модификацию данных

INSERT

```
INSERT INTO opt.order_item (order_item_id, product_id, order_id, unit_price, quantity_in_order)
SELECT
      'FEE_' || co.order_id AS order_item_id,
      'P_FEE' AS product_id,
      co.order_id AS order_id,
```

```

50 AS unit_price,
1 AS quantity_in_order
FROM opt.customer_order co
WHERE co.payment_status = 'Paid'
AND co.order_id IN (
SELECT oi.order_id
FROM opt.order_item oi
GROUP BY oi.order_id
HAVING SUM(oi.unit_price * oi.quantity_in_order) > 1000
)
AND NOT EXISTS (
SELECT 1
FROM opt.order_item oi_check
WHERE oi_check.order_id = co.order_id AND oi_check.product_id = 'P_FEE'
);

```

До

```

SELECT oi.order_item_id, oi.product_id, oi.order_id
FROM opt.order_item oi
WHERE oi.product_id = 'P_FEE'
AND oi.order_id IN ('0002', '0004');

```

Data Output	Messages	Notifications
<div> <div>SQL</div> </div>		
customer_id [PK] character varying (10)	name character varying (20)	address character varying (50)

После

Data Output

Messages

Notifications

## UPDATE

```

UPDATE opt.order_item oi
SET unit_price = ROUND(oi.unit_price * 1.15)
FROM opt.customer_order co
WHERE oi.order_id = co.order_id
AND co.payment_status = 'Pending'
AND oi.product_id IN (
SELECT DISTINCT di.product_id
FROM opt.delivery_item di
JOIN opt.delivery d ON di.delivery_id = d.delivery_id
WHERE d.warehouse_id = 'W001'
);

```

До



```

SELECT oi.order_item_id, oi.order_id, oi.product_id, oi.unit_price
FROM opt.order_item oi
WHERE oi.order_id IN ('0001', '0003', '0005')
AND oi.product_id IN (
    SELECT DISTINCT di.product_id
    FROM opt.delivery_item di
    JOIN opt.delivery d ON di.delivery_id = d.delivery_id
    WHERE d.warehouse_id = 'W001'
);

```

	order_item_id [PK] character varying (10)	order_id character varying (10)	product_id character varying (10)	unit_price integer
1	OI001	O001	P001	350
2	OI003	O003	P003	25
3	OI006	O005	P006	3
4	OI010	O005	P002	1100
5	OI004	O003	P004	55

После

Data Output Messages Notifications				
	order_item_id [PK] character varying (10)	order_id character varying (10)	product_id character varying (10)	unit_price integer
1	OI001	O001	P001	403
2	OI003	O003	P003	29
3	OI006	O005	P006	3
4	OI010	O005	P002	1265
5	OI004	O003	P004	63

## DELETE

```

DELETE
FROM opt.order_item oi USING opt.product p,
    opt.customer_order co,
    opt.manager m,
    opt.position_assignment pa,
    opt."position" pos
WHERE oi.product_id = p.product_id
AND oi.order_id = co.order_id
AND co.manager_id = m.manager_id
AND m.manager_id = pa.manager_id
AND pa.position_id = pos.position_id
AND p.manufacturer_id = 'MF4'
AND pos.title = 'Sales Manager'
AND pa.end_date IS NULL;

```

До

Data Output Messages Notifications						
	order_item_id character varying (10)	order_id character varying (10)	product_id character varying (10)	product_name character varying (20)	manager_id character varying (10)	manager_position character varying (20)
1	OI004	O003	P004	Juice 1L	MGR3	Sales Manager

```
SELECT oi.order_item_id, oi.order_id, oi.product_id, p.name as product_name, m.manager_id,
pos.title as manager_position
FROM opt.order_item oi
JOIN opt.product p ON oi.product_id = p.product_id
JOIN opt.customer_order co ON oi.order_id = co.order_id
JOIN opt.manager m ON co.manager_id = m.manager_id
JOIN opt.position_assignment pa ON m.manager_id = pa.manager_id
JOIN opt."position" pos ON pa.position_id = pos.position_id
WHERE p.manufacturer_id = 'MF4'
AND pos.title = 'Sales Manager'
AND pa.end_date IS NULL;
```

После

Data Output Messages Notifications						
	order_item_id character varying (10)	order_id character varying (10)	product_id character varying (10)	product_name character varying (20)	manager_id character varying (10)	manager_position character varying (20)

## Индексы

Запрос: Найдем все заказы клиента CUST1.

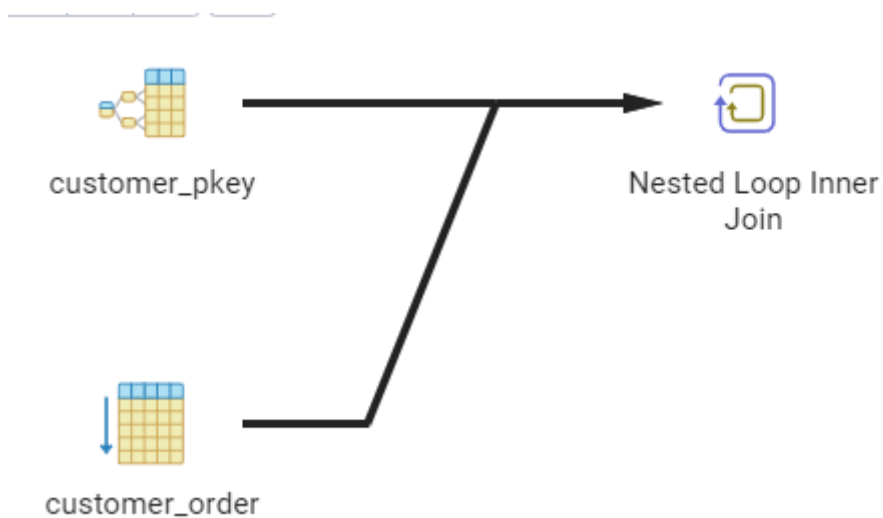
```
SELECT co.order_id, co.order_status, co.payment_status, co.order_date, c.name
AS customer_name
FROM opt.customer_order co
JOIN opt.customer c ON co.customer_id = c.customer_id
WHERE co.customer_id = 'CUST1';
```

Без индексов:

Время:

Successfully run. Total query runtime: 63 msec.

План:



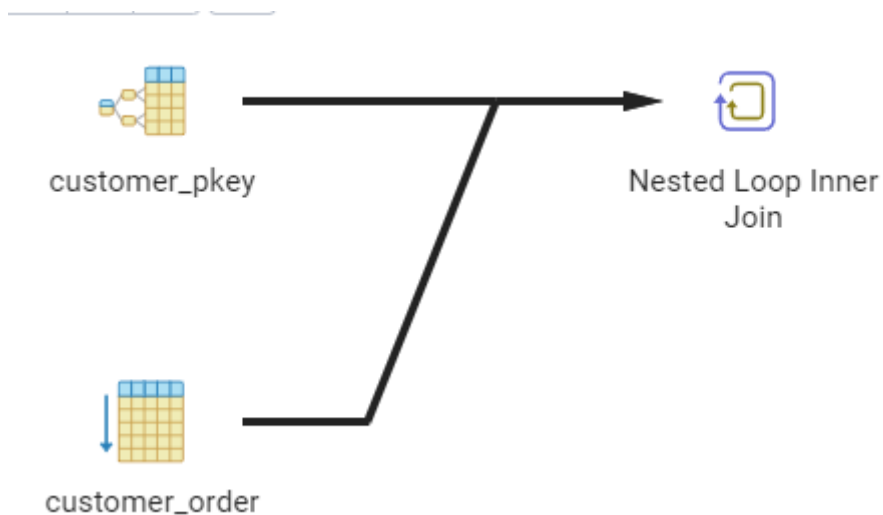
Индекс:

```
CREATE INDEX idx_co_customer_id ON opt.customer_order(customer_id);
```

Время:

Successfully run. Total query runtime: 47 msec.

План:



Запрос: Найдем все позиции заказов для товара P001, где количество в заказе больше 1.

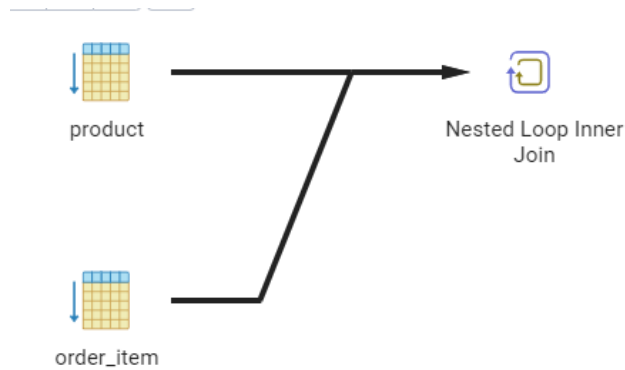
```
SELECT oi.order_item_id, oi.order_id, p.name AS product_name, oi.quantity_in_order, oi.unit_price
FROM opt.order_item oi
      JOIN opt.product p ON oi.product_id = p.product_id
WHERE oi.product_id = 'P001'
      AND oi.quantity_in_order > 1;
```

Без индексов:

Время:

Successfully run. Total query runtime: 75 msec.

План:



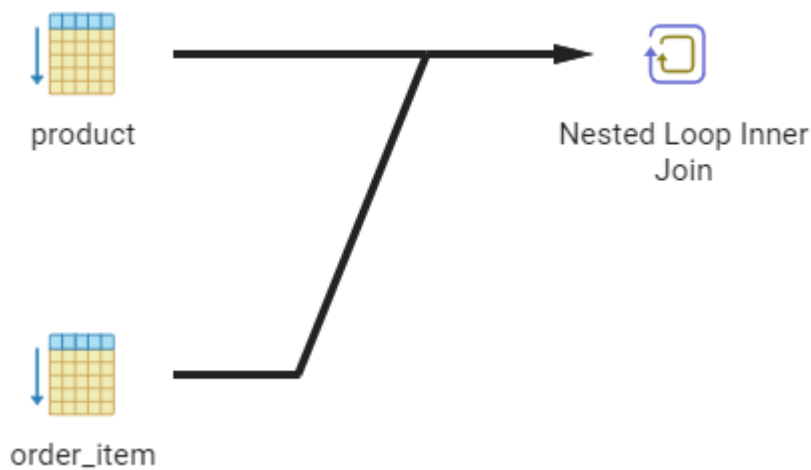
Индекс:

```
CREATE INDEX idx_oi_product_quantity ON opt.order_item(product_id, quantity_in_order);
```

Время:

Successfully run. Total query runtime: 53 msec.

План:



### Вывод

В ходе выполнения лабораторной работы были освоены основные операции по работе с данными в СУБД PostgreSQL с использованием Query Tool, включая их выборку, добавление, обновление и удаление. Также были получены навыки создания представлений для удобного доступа к информации и создания индексов (простых и составных) для полей таблиц. При этом на практике с помощью EXPLAIN ANALYZE было проанализировано, как индексы могут существенно повысить скорость выполнения запросов и оптимизировать работу базы данных.