

Национальный исследовательский Университет ИТМО  
Мегафакультет информационных и трансляционных технологий  
Факультет инфокоммуникационных технологий

## Базы данных

# "Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL"

Лабораторная работа №2

**Работу**

**выполнил:**

Нестеров В.А.

Группа: К3243

**Преподаватель:**

Говорова М.М.

Санкт-Петербург

2022

# Содержание

<b>1. Цель работы</b>	<b>3</b>
<b>2. Практическое задание</b>	<b>3</b>
<b>3. Схема базы данных</b>	<b>3</b>
<b>4. Выполнение</b>	<b>4</b>
4.1. Запросы к базе данных. . . . .	4
4.1.1. Вывести список поставщиков, которые поставляют все товары. . .	4
4.1.2. Определить поставщика, который поставляет каждый из товаров по самой низкой цене. . . . .	4
4.1.3. Вывести названия товаров, цены на которые у всех поставщиков одинаковы. . . . .	5
4.1.4. Чему равен общий суточный доход оптового склада за прошедший день? . . . . .	5
4.1.5. Вычислить стоимость каждого вида товара, находящегося на базе.	5
4.1.6. В какой день было вывезено минимальное количество товара? . . .	6
4.1.7. Сколько различных видов товара имеется на базе? . . . . .	6
4.2. Представления. . . . .	7
4.2.1. Фирмы-покупатели . . . . .	7
4.2.2. Доход базы . . . . .	7
4.3. Запросы на модификацию данных. . . . .	8
4.3.1. INSERT . . . . .	8
4.3.2. UPDATE . . . . .	9
4.3.3. DELETE . . . . .	9
4.4. Создание индексов. . . . .	10
4.4.1. Количество поставок, доставленных на Базу в течение прошедшего месяца. . . . .	10
<b>5. Выводы</b>	<b>11</b>
<b>Заключение</b>	<b>11</b>

# 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

- 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.

## 3. Схема базы данных

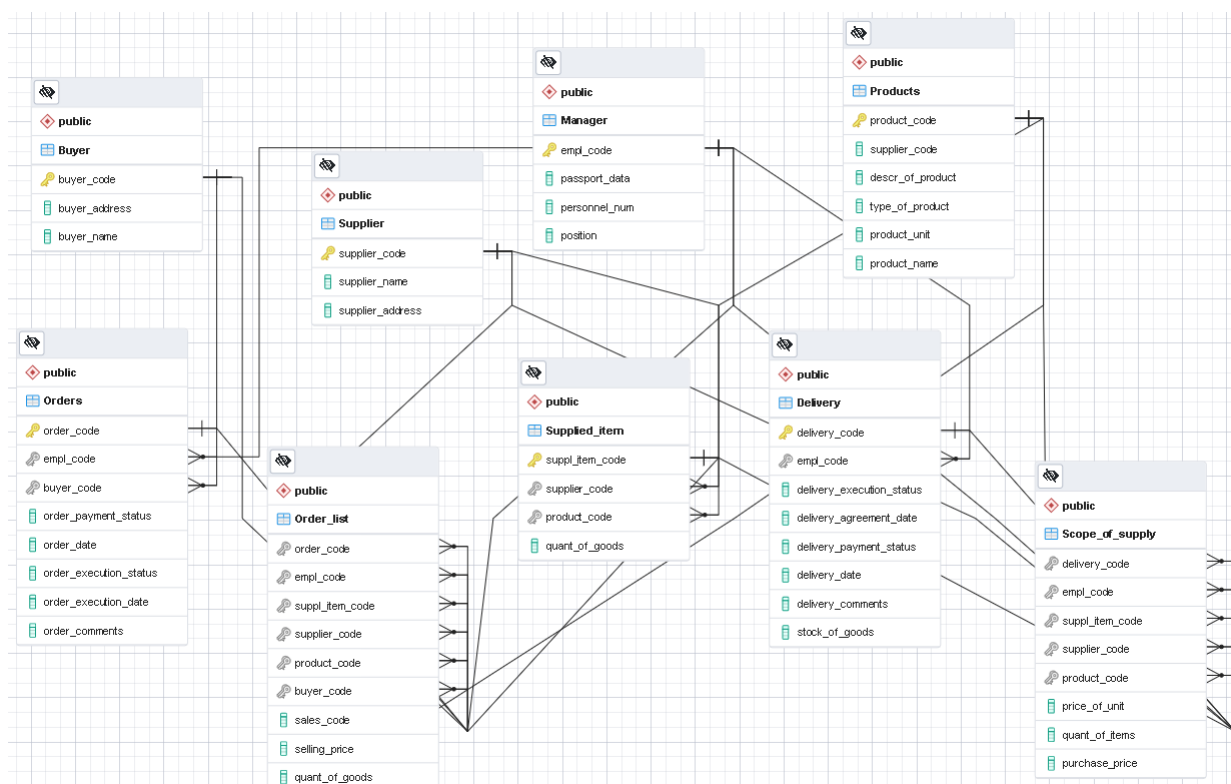


Рисунок 3.1. Модель базы данных

## 4. Выполнение

### 4.1. Запросы к базе данных.

#### 4.1.1. Вывести список поставщиков, которые поставляют все товары.

```
SELECT DISTINCT "Supplier".SUPPLIER_CODE,  
SUPPLIER_NAME  
FROM "Supplier"  
JOIN "Products" ON "Supplier".SUPPLIER_CODE = "Products".SUPPLIER_CODE
```

	supplier_code [PK] character varying	supplier_name character varying
88	43	Vestibulum Lorem Limited
89	75	Erat Eget LLC
90	17	Natoque Penatibus Corporation
91	79	Dolor Dapibus Incorporated
92	18	Nunc Company
93	10	Mauris PC
94	81	Felis Ullamcorper Associates
95	64	Erat Eget Limited
96	19	Massa Rutrum LLC
97	97	Libero Proin Industries
98	6	Scelerisque Limited
99	87	Dolor Sit Ltd
100	67	Consequat Enim Diam Limited

Рисунок 4.1. Список поставщиков, поставляющих все товары

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

```
SELECT "table1".SUPPLIER_CODE,  
"table1".PRODUCT_CODE,  
"table1".PURCHASE_PRICE  
FROM  
(SELECT DISTINCT PRODUCT_CODE,  
SUPPLIER_CODE,  
PURCHASE_PRICE  
FROM "Scope_of_supply") AS TABLE1
```

	supplier_code character varying	product_code character varying	purchase_price integer
1	51	51	8901
2	29	29	8434
3	25	25	7686
4	22	22	3759
5	60	60	69
6	93	93	6014
7	61	61	168
8	48	48	3370
9	1	1	1313

Рисунок 4.2. Определить поставщика

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

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

```
select SUM (selling_price * "Order_list".quant_of_goods)-("Scope_of_supply".quant_of_goods)
from "Order_list"
LEFT JOIN "Scope_of_supply" ON "Scope_of_supply".empl_code = "Order_list".empl_code
LEFT JOIN "Orders" ON "Orders".order_code = "Order_list".order_code
WHERE "Orders".order_date ≥ current_date - interval '1 day'
and "Orders".order_date < current_date
group by "Scope_of_supply".purchase_price, "Scope_of_supply".quant_of_items
```

	?column? bigint
1	-23631

Рисунок 4.3. Общий суточный доход за прошедший день

**4.1.5. Вычислить стоимость каждого вида товара, находящегося на базе.**

```
SELECT DISTINCT TYPE_OF_PRODUCT,
PRICE_OF_UNIT
FROM
(SELECT *
FROM "Scope_of_supply"
LEFT JOIN "Products" ON "Scope_of_supply".PRODUCT_CODE = "Products".PRODUCT_CODE) AS FOO
```

	type_of_product character varying	price_of_unit integer
1	purus. Duis	28562
2	mattis ornare,	20634
3	Donec nibh	17341
4	luctus et	11390
5	iaculis quis,	76275
6	iaculis quis,	30252
7	Curabitur vel	48073
8	elementum, lorem	81091
9	aliquet nec,	80109
10	auctor. Mauris	20261
11	in sodales	56021
12	nonummy ipsum	17409
13	non, cursus	10459
14	augue. Sed	96385
15	vel, ipsum et	87658

Рисунок 4.4. Стоимость каждого вида товара, находящегося на базе

#### 4.1.6. В какой день было вывезено минимальное количество товара?

```
select order_execution_date, quant_of_goods from
(select "Orders".order_code, order_execution_date, quant_of_goods from "Orders"
left join "Order_list" on "Order_list".order_code = "Orders".order_code) as foo
where quant_of_goods = (select min(quant_of_goods) from "Order_list")
```

	order_execution_date date	quant_of_goods integer
1	2022-06-19	1
2	2023-05-09	1
3	2022-10-11	1
4	2021-11-05	1

Рисунок 4.5. День, в который было вывезено минимальное количество товара

#### 4.1.7. Сколько различных видов товара имеется на базе?

```
SELECT COUNT (*)
FROM
(SELECT DISTINCT TYPE_OF_PRODUCT
FROM "Products"
LEFT JOIN "Scope_of_supply" ON "Products".PRODUCT_CODE = "Scope_of_supply".PRODUCT_CODE
LEFT JOIN "Delivery" ON "Scope_of_supply".DELIVERY_CODE = "Delivery".DELIVERY_CODE
WHERE (DELIVERY_DATE ≤ 'today')
AND (DELIVERY_EXECUTION_STATUS = 'Yes')) AS DISTINCT_TYPES_OF_PRODUCT
```

	count	
	bigint	
1		26

Рисунок 4.6. Количество различных видов товара на Базе

## 4.2. Представления.

### 4.2.1. Фирмы-покупатели

Количество заказов фирм-покупателей за прошедший год.

```
SELECT count(*)
FROM "Orders"
WHERE ((Order_date ≥ '2021-01-01') and (Order_date ≤ '2022-12-12'))
```

	count	
	bigint	
1		68

Рисунок 4.7. Скриншот выполнения запроса



Рисунок 4.8. Скриншот представления

### 4.2.2. Доход базы

Доход базы за конкретный период.

```
SELECT SUM(SELLING_PRICE * QUANT_OF_GOODS)
FROM
(SELECT "Orders".ORDER_CODE,
SELLING_PRICE,
QUANT_OF_GOODS,
"Orders".ORDER_DATE
FROM "Order_list"
LEFT JOIN "Orders" ON "Order_list".ORDER_CODE = "Orders".ORDER_CODE) AS FOO
WHERE ((ORDER_DATE ≥ '2022-01-01')
AND (ORDER_DATE ≤ '2022-05-01'))
```

	sum	
	bigint	
1	292974	

Рисунок 4.9. Скриншот выполнения запроса

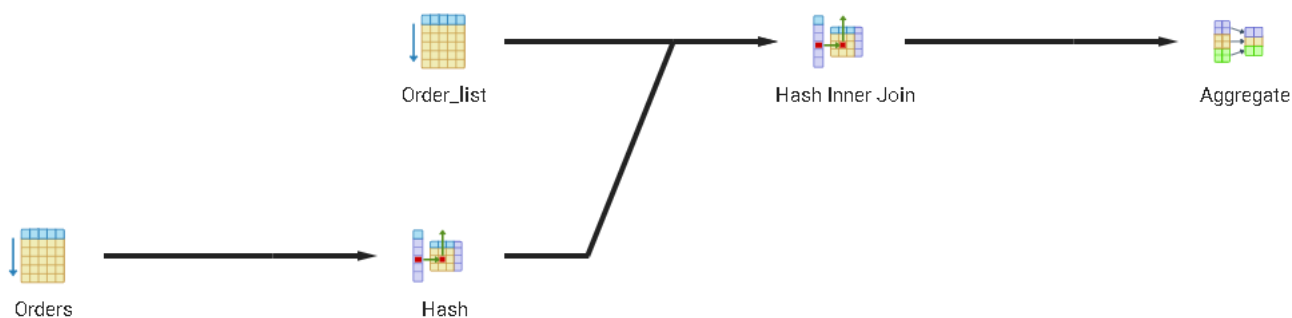


Рисунок 4.10. Скриншот представления

## 4.3. Запросы на модификацию данных.

### 4.3.1. INSERT

Добавим заказ в таблицу "Orders".

```

INSERT INTO public."Orders"(
order_code, empl_code, buyer_code, order_payment_status, order_date, order_execution_status,
VALUES (101, 100, 100, 'Yes', '2022-06-28', 'Yes', '2022-06-28', '');
  
```

	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
88	88	88	88	No	2022-04-01	No	2022-01-16	malesuada ut, sem. Nulla interdum. Curabitur
89	89	89	89	No	2022-03-19	Yes	2023-02-07	Phasellus elit pede, malesuada vel, venenatis vel,
90	90	90	90	Yes	2022-04-09	Yes	2021-06-30	tincidunt,
91	91	91	91	No	2022-04-10	No	2022-07-23	et, lacinia vitae, sodales at,
92	92	92	92	No	2022-09-08	No	2022-08-31	sodales nisi magna sed dui. Fusce aliquam, enim nec temp
93	93	93	93	No	2023-01-17	Yes	2022-12-17	mus. Proin vel nisl. Quisque fringilla euismod enim. Etiam
94	94	94	94	Yes	2022-03-31	No	2023-01-26	pulvinar arcu et pede. Nunc sed orci lobortis augue soeleris
95	95	95	95	Yes	2022-11-06	Yes	2021-08-25	nec, mollis
96	96	96	96	Yes	2022-11-03	No	2022-10-25	eu neque pellentesque massa lobortis ultrices.
97	97	97	97	No	2022-05-08	No	2021-06-26	velit. Quisque varius.
98	98	98	98	Yes	2022-07-21	No	2022-06-30	sem eget massa. Suspendisse eleifend. Cras sed leo. Cras
99	99	99	99	Yes	2023-03-27	Yes	2022-05-31	mollis. Phasellus
100	100	100	100	No	2022-06-10	Yes	2022-08-31	vel arcu eu odio tristique pharetra. Quisque ac libero nec

Рисунок 4.11. До добавления в таблицу



	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
89	89	89	89	No	2022-03-19	Yes	2023-02-07	Phasellus elit pede, malesuada vel, venenatis vel,
90	90	90	90	Yes	2022-04-09	Yes	2021-06-30	tincidunt,
91	91	91	91	No	2022-04-10	No	2022-07-23	et, lacinia vitae, sodales at,
92	92	92	92	No	2022-09-08	No	2022-08-31	sodales nisi magna sed dui. Fusce aliquam, enim nec ten
93	93	93	93	No	2023-01-17	Yes	2022-12-17	mus. Proin vel nisl. Quisque fringilla euismod enim. Etia
94	94	94	94	Yes	2022-03-31	No	2023-01-26	pulvinar arcu et pede. Nunc sed orci lobortis augue soele
95	95	95	95	Yes	2022-11-06	Yes	2021-08-25	nec, mollis
96	96	96	96	Yes	2022-11-03	No	2022-10-25	eu neque pellentesque massa lobortis ultricies.
97	97	97	97	No	2022-05-08	No	2021-06-26	velit. Quisque varius.
98	98	98	98	Yes	2022-07-21	No	2022-06-30	sem eget massa. Suspendisse eleifend. Cras sed leo. Cra
99	99	99	99	Yes	2023-03-27	Yes	2022-05-31	mollis. Phasellus
100	100	100	100	No	2022-06-10	Yes	2022-08-31	vel arcu eu odio tristique pharetra. Quisque ac libero nec
101	101	100	100	Yes	2022-06-28	Yes	2022-06-28	

Рисунок 4.12. После добавления в таблицу

### 4.3.2. UPDATE

Обновим статусы заказов от 26.06.2022 до 02.07.2022 на доставленные.

```
update "Orders"
set order_execution_status = 'Yes'
where order_execution_date ≥ '2022-06-26' and order_execution_date ≤ '2022-07-02'
```

	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
1	26	26	26	No	2023-04-18	No	2022-07-02	porttitor soelerisque neque.
2	98	98	98	Yes	2022-07-21	No	2022-06-30	sem eget massa. Suspendisse eleifend. Cras sed leo. Cras v
3	101	100	100	Yes	2022-06-28	Yes	2022-06-28	

Рисунок 4.13. До изменения таблицы

	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
1	26	26	26	No	2023-04-18	Yes	2022-07-02	porttitor soelerisque neque.
2	98	98	98	Yes	2022-07-21	Yes	2022-06-30	sem eget massa. Suspendisse eleifend. Cras sed leo. Cras v
3	101	100	100	Yes	2022-06-28	Yes	2022-06-28	

Рисунок 4.14. После изменения таблицы

### 4.3.3. DELETE

Удалим заказ с кодом 101.

```
delete from "Orders"
where order_code = '101'
```

	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
1	101	100	100	Yes	2022-06-28	Yes	2022-06-28	

Рисунок 4.15. До удаления заказа

	order_code [PK] character varying	empl_code character varying	buyer_code character varying	order_payment_status character varying	order_date date	order_execution_status character varying	order_execution_date date	order_comments character varying
--	--------------------------------------	--------------------------------	---------------------------------	---	--------------------	---	------------------------------	-------------------------------------

Рисунок 4.16. После удаления заказа

## 4.4. Создание индексов.

### 4.4.1. Количество поставок, доставленных на Базу в течение прошедшего месяца.

```
SELECT COUNT (*)
FROM
(SELECT DELIVERY_CODE,
DELIVERY_DATE
FROM "Delivery"
WHERE (DELIVERY_DATE ≥ CURRENT_DATE - interval '30 days')
AND (DELIVERY_DATE ≤ 'today')) AS F00
```

	QUERY PLAN	
	text	
1	Aggregate (cost=4.01..4.02 rows=1 width=8)	
2	[...] -> Seq Scan on "Delivery" (cost=0.00..4.00 rows=6 width=0)	
3	[...] Filter: ((delivery_date <= '2022-06-30'::date) AND (delivery_date >= (CURRENT_DATE - '30 days'::interval)))	

Рисунок 4.17. План выполнения запроса до создания индекса

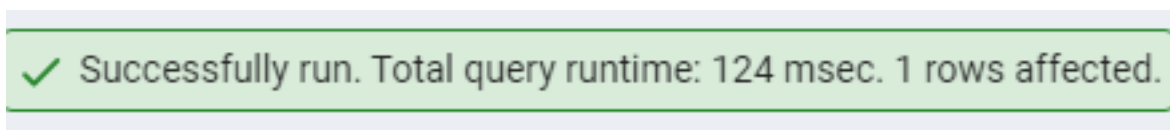


Рисунок 4.18. Время выполнения запроса до создания индекса

### Создадим индекс и выполним запрос.

	QUERY PLAN	
	text	
1	Aggregate (cost=4.01..4.02 rows=1 width=8)	
2	[...] -> Seq Scan on "Delivery" (cost=0.00..4.00 rows=6 width=0)	
3	[...] Filter: ((delivery_date <= '2022-06-30'::date) AND (delivery_date >= (CURRENT_DATE - '30 days'::interval)))	

Рисунок 4.19. План выполнения запроса после создания индекса

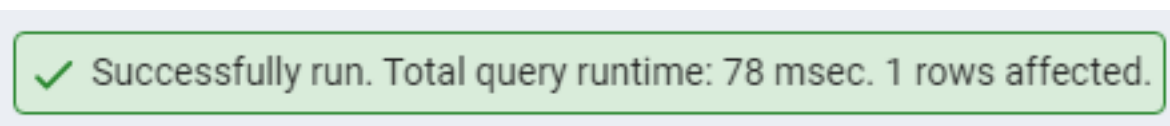


Рисунок 4.20. Время выполнения запроса после создания индекса

### Удалим индекс

```
DROP INDEX IF EXISTS month_deliveries
```

## 5. Выводы

В данной работе мы овладели практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.