

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

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ № 4

по теме: запросы на выборку данных к БД PostgreSQL. Представления в PostgreSQL.

по дисциплине: Проектирование и реализация баз данных

Проверил:

Горова М.М. _____

Дата: «27» мая 2021г.

Оценка _____

Выполнил:

студент группы К3243

Измайлова Анна

Санкт-Петербург 2020

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL и использования подзапросов при модификации данных.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4. **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.
4. Просмотреть историю запросов.

Индивидуальное задание:

Вариант No9 “Оптовая база”

1. Название БД

«Wholesale base»

2. Схема инфологической модели данных БД

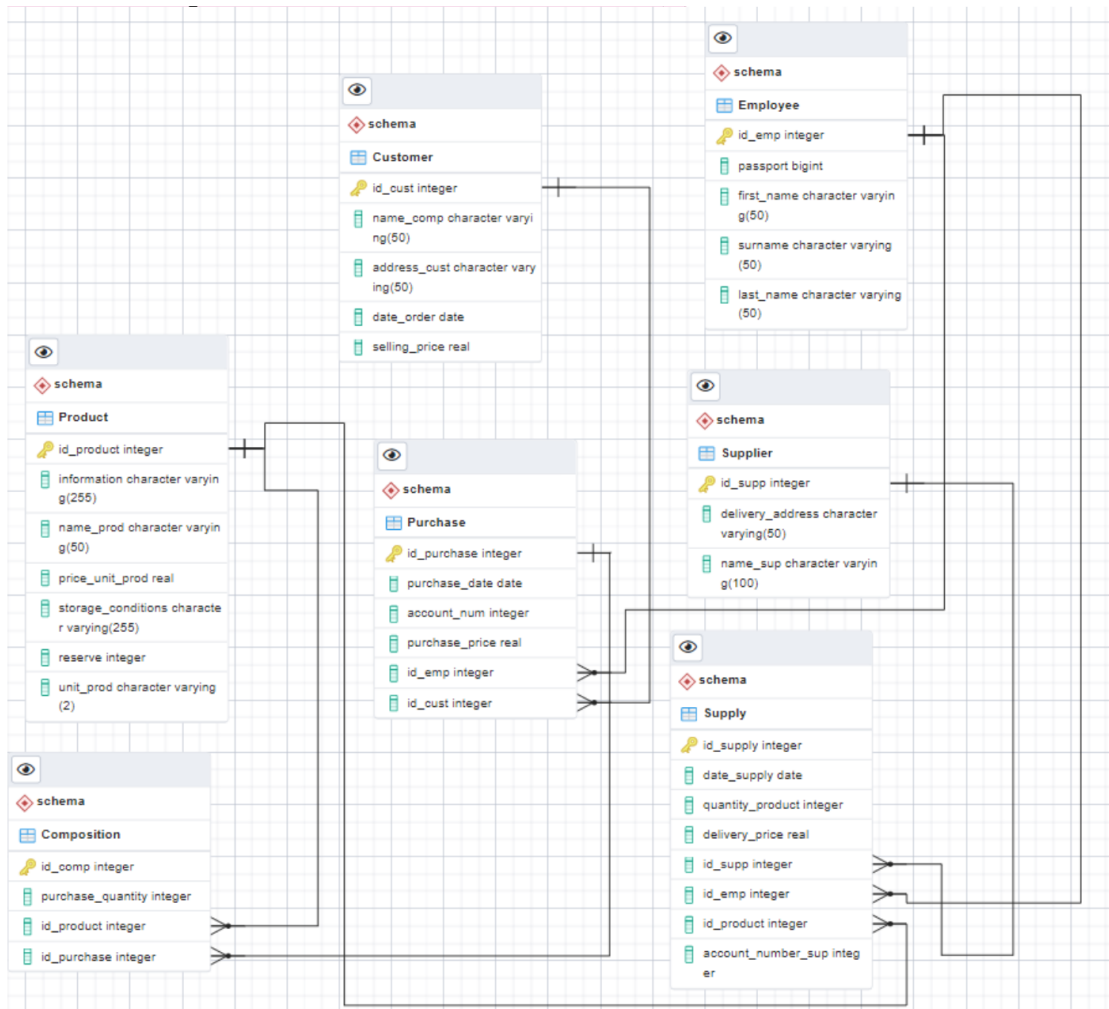


Рисунок 1 – Схема инфологической модели БД, сгенерированная в Generate ERD

3. Запросы к базе данных

- Вывести список поставщиков, которые поставляют все товары.

Wholesale base/postgres@PostgreSQL 11							
Query Editor	История запросов						
<pre>1 SELECT schema."Supplier".name_sup FROM 2 (SELECT id_supp FROM (SELECT id_supp, COUNT (DISTINCT id_product)AS count_products 3 FROM schema."Supply" GROUP BY id_supp) AS tab 4 WHERE (count_products = (SELECT COUNT(id_product) FROM schema."Product"))AS tab1 5 INNER JOIN schema."Supplier" ON tab1.id_supp = schema."Supplier".id_supp;</pre>							
Результат	План выполнения Сообщения Notifications						
<table><thead><tr><th></th><th>name_sup</th></tr><tr><th></th><th>character varying (100)</th></tr></thead><tbody><tr><td>1</td><td>Аврора</td></tr></tbody></table>			name_sup		character varying (100)	1	Аврора
	name_sup						
	character varying (100)						
1	Аврора						


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

Wholesale base/postgres@PostgreSQL 11							
Query Editor	История запросов						
<pre>1 SELECT schema."Supplier".name_sup FROM 2 (SELECT id_supp, COUNT(id_product) 3 FROM (SELECT id_product, id_supp, MIN(delivery_price) AS min_price 4 FROM schema."Supply" GROUP BY id_product, id_supp)as foo GROUP BY id_supp 5 HAVING COUNT(id_product) = (SELECT COUNT(DISTINCT id_product) FROM schema."Supply")) AS tab 6 INNER JOIN schema."Supplier" ON tab.id_supp = schema."Supplier".id_supp;</pre>							
Результат	План выполнения Сообщения Notifications						
<table><thead><tr><th></th><th>name_sup</th></tr><tr><th></th><th>character varying (100)</th></tr></thead><tbody><tr><td>1</td><td>Аврора</td></tr></tbody></table>			name_sup		character varying (100)	1	Аврора
	name_sup						
	character varying (100)						
1	Аврора						



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

Wholesale base/postgres@PostgreSQL 11															
Query Editor	История запросов														
<pre>1 SELECT schema."Product".name_prod FROM 2 (SELECT id_product, COUNT(DISTINCT delivery_price) AS sum FROM schema."Supply" GROUP BY id_product) f_result, schema."Product" 3 where schema."Product".id_product = f_result.id_product</pre>															
Результат	План выполнения Сообщения Notifications														
<table><thead><tr><th></th><th>name_prod</th></tr><tr><th></th><th>character varying (50)</th></tr></thead><tbody><tr><td>1</td><td>горбыль</td></tr><tr><td>2</td><td>необрезная доска</td></tr><tr><td>3</td><td>обрезная доска</td></tr><tr><td>4</td><td>четвертина</td></tr><tr><td>5</td><td>пластина</td></tr></tbody></table>			name_prod		character varying (50)	1	горбыль	2	необрезная доска	3	обрезная доска	4	четвертина	5	пластина
	name_prod														
	character varying (50)														
1	горбыль														
2	необрезная доска														
3	обрезная доска														
4	четвертина														
5	пластина														



-Чему равен общий суточный доход оптового склада за прошедший день

Wholesale base/postgres@PostgreSQL 11 ▾	
Query Editor	История запросов
<pre>1 SELECT SUM(purchase_price) AS result 2 FROM schema."Purchase" 3 WHERE purchase_date = TIMESTAMP 'yesterday';</pre>	
Результат	План выполнения Сообщения Notifications
 result real	
1	55465

-Вычислить стоимость товаров, оставшихся на складе

Wholesale base/postgres@PostgreSQL 11 ▾	
Query Editor	История запросов
<pre>1 SELECT name_prod, price_unit_prod*reserve AS result 2 FROM schema."Product" 3 WHERE reserve > 0;</pre>	
Результат	План выполнения Сообщения Notifications
 name_prod character varying (50)	 result double precision
1	пластина 150000
2	четвертина 200000
3	горбыль 250000
4	обрезная доска 230000
5	необрезная доска 300000

-В какой день было вывезено минимальное количество товара

Wholesale base/postgres@PostgreSQL 11 ▾	
Query Editor	История запросов
<pre>1 SELECT date_supply, quantity_product 2 FROM schema."Supply" 3 WHERE quantity_product = (SELECT min(quantity_product) FROM schema."Supply");</pre>	
Результат	План выполнения Сообщения Notifications
 date_supply date	 quantity_product integer
1	2020-01-15 10
2	2020-01-15 10
3	2020-01-15 10
4	2020-01-16 10

Сколько различных видов товара имеется на базе

Wholesale base/postgres@PostgreSQL 11

Query Editor

```
1 SELECT SUM(qty) AS quantity
2 FROM (SELECT COUNT(DISTINCT name_prod) AS qty FROM schema."Product" GROUP BY id_product) AS tab;
```

Query Editor История запросов План выполнения

Результат Сообщения Notifications

quantity	
numeric	
1	5

4. Представления

1) Количество заказов фирм-покупателей за прошедший год

Wholesale base/postgres@PostgreSQL 11

Query Editor История запросов

```
1 CREATE VIEW schema.comp AS
2 SELECT sum(count_purchase) AS quantity_purchase
3 FROM (SELECT schema."Customer".name_comp, tab.count_purchase
4 FROM (SELECT id_cust, COUNT(id_cust) AS count_purchase
5 FROM schema."Purchase"
6 WHERE purchase_date BETWEEN '2020-01-01' AND '2020-12-31'
7 GROUP BY id_cust) as tab
8 INNER JOIN schema."Customer" ON tab.id_cust = schema."Customer".id_cust) as result;
```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно. время выполнения: 121 msec.

Вид:

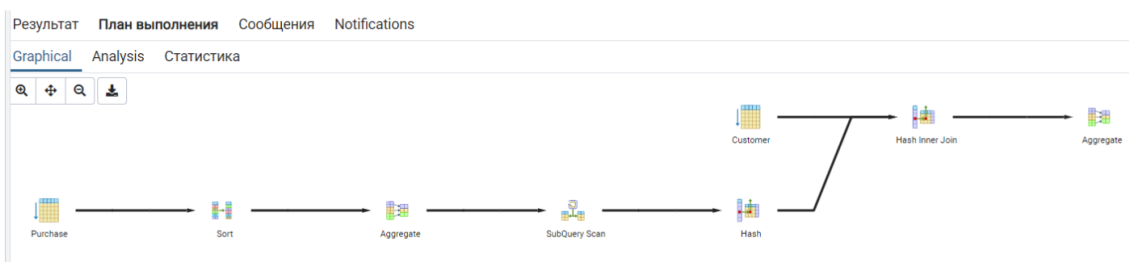
Wholesale base/postgres@PostgreSQL 11

Query Editor История запросов

```
1 SELECT quantity_purchase
2 FROM schema.comp;
```

Результат План выполнения Сообщения Notifications

quantity_purchase	
numeric	
1	8



2) Доход базы за конкретный период.

Wholesale base/postgres@PostgreSQL 11 ▾

Query Editor История запросов

```
1 CREATE VIEW schema.warehouse AS
2 SELECT sum(purchase_price) AS proceeds
3 FROM schema."Purchase"
4 where purchase_date BETWEEN '2020-01-01' AND '2020-12-31';
```

Результат План выполнения Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 93 мсек.

Вид:

Wholesale base/postgres@PostgreSQL 11 ▾

Query Editor История запросов

```
1 SELECT proceeds
2 FROM schema.warehouse;
```

Результат План выполнения Сообщения Notifications

	proceeds real
1	69145

Wholesale base/postgres@PostgreSQL 11 ▾

Query Editor История запросов

```
1 SELECT proceeds
2 FROM schema.warehouse;
```

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика

🔍 🔄 🔍 📄

Purchase → Aggregate

5. Запросы на модификацию данных

1) INSERT

Добавляем новую запись о поставке для продуктов с ценой ниже 200.

Wholesale base/postgres@PostgreSQL 11 ▾

Query Editor История запросов

```
1 insert into schema."Supply"
2 SELECT (select max(id_supply)+1 from schema."Supply"), '2020-05-21', 10, 100, 2, 2, id_product, 43241
3 FROM schema."Product"
4 WHERE price_unit_prod < 200;
```

Результат План выполнения Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 112 мсек.

До:

	id_supply [PK] integer	date_supply date	quantity_product integer	delivery_price real	id_supp integer	id_emp integer	id_product integer	account_number_sup integer
1	8	2020-01-15	10	100	2	2	5	67766
2	7	2020-01-15	10	110	2	2	4	46744
3	6	2020-01-16	10	500.5	2	2	3	38974
4	4	2020-10-01	34	130	1	4	1	40123
5	2	2020-01-15	30	590	2	2	2	20123
6	3	2020-01-15	32	550	3	3	3	30123
7	9	2020-02-13	13	480	2	3	3	13432
8	1	2020-01-10	30	490	2	1	3	10123
9	10	2020-02-02	12	90	2	1	1	32343
10	11	2020-02-12	12	54	1	3	2	32422

После:

	id_supply [PK] integer	date_supply date	quantity_product integer	delivery_price real	id_supp integer	id_emp integer	id_product integer	account_number_sup integer
1	8	2020-01-15	10	100	2	2	5	67766
2	7	2020-01-15	10	110	2	2	4	46744
3	6	2020-01-16	10	500.5	2	2	3	38974
4	4	2020-10-01	34	130	1	4	1	40123
5	2	2020-01-15	30	590	2	2	2	20123
6	3	2020-01-15	32	550	3	3	3	30123
7	9	2020-02-13	13	480	2	3	3	13432
8	1	2020-01-10	30	490	2	1	3	10123
9	10	2020-02-02	12	90	2	1	1	32343
10	11	2020-02-12	12	54	1	3	2	32422
11	12	2020-05-21	10	100	2	2	1	43241

2) UPDATE

Обновление цены покупки, которая была до 2020-02-15

Wholesale base/postgres@PostgreSQL 11

Query Editor
История запросов

```

1 delete from schema."Customer"
2 where id_cust in
3 (select id_cust from schema."Purchase"
4  where schema."Purchase".purchase_price < 6000);

```

Результат
План выполнения
Сообщения
Notifications

DELETE 2

Запрос завершён успешно, время выполнения: 145 мсек.

До:

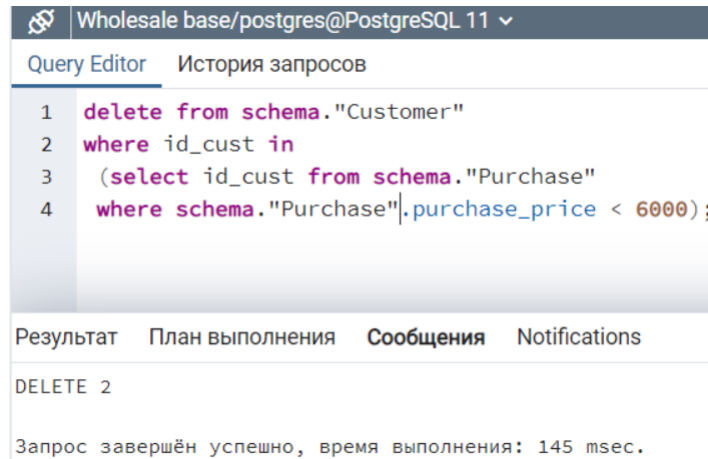
	id_purchase [PK] integer	purchase_date date	account_num integer	purchase_price real	id_emp integer	id_cust integer
1	1	2020-02-28	10123	4500	1	1
2	2	2020-03-02	20123	6000	2	2
3	3	2020-04-15	30123	8000	3	3
4	4	2020-02-28	12324	9500	3	2
5	5	2020-02-28	42314	23456	3	3
6	6	2020-02-28	41325	12342	2	3
7	7	2021-05-11	31241	43123	1	2
8	8	2021-05-11	31234	12342	2	2
9	9	2020-02-12	32143	2134	2	2
10	10	2020-02-02	32142	3213	1	1

После:

	id_purchase [PK] integer	purchase_date date	account_num integer	purchase_price real	id_emp integer	id_cust integer
1	2	2020-03-02	20123	6000	2	2
2	3	2020-04-15	30123	8000	3	3
3	4	2020-02-28	12324	9500	3	2
4	5	2020-02-28	42314	23456	3	3
5	6	2020-02-28	41325	12342	2	3
6	7	2021-05-11	31241	43123	1	2
7	8	2021-05-11	31234	12342	2	2
8	9	2020-02-12	32143	2134	2	2
9	1	2020-02-28	10123	3000	1	1
10	10	2020-02-02	32142	3000	1	1

3) DELETE

Удаление покупателей, у которых цена покупки была меньше 6000.



The screenshot shows a PostgreSQL Query Editor window titled "Wholesale base/postgres@PostgreSQL 11". The "Query Editor" tab is active, displaying the following SQL query:

```
1 delete from schema."Customer"  
2 where id_cust in  
3 (select id_cust from schema."Purchase"  
4 where schema."Purchase".purchase_price < 6000);
```

Below the query editor, the "Результат" (Result) tab is selected, showing the output of the query:

```
DELETE 2
```

A message at the bottom states: "Запрос завершён успешно, время выполнения: 145 msec."

До:

Результат					
	id_cust [PK] integer	name_comp character varying (50)	address_cust character varying (50)	date_order date	selling_price real
1	1	ИЛИМ	г. Ярославль	2020-02-12	11000
2	2	Титан	г. Кострома	2020-02-15	11500
3	3	УЛК	г. Вологда	2020-04-01	12500

После:

Результат					
	id_cust [PK] integer	name_comp character varying (50)	address_cust character varying (50)	date_order date	selling_price real
1	3	УЛК	г. Вологда	2020-04-01	12500

Вывод:

Созданы запросы и представления на выборку данных к базе данных Wholesale base (согласно индивидуальному заданию, часть 2 и 3). Составлены 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Изучить графическое представление запросов. Просмотреть историю запросов.