

Министерство науки и высшего образования Российской  
Федерации Федеральное государственное автономное  
образовательное учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»  
Факультет инфокоммуникационных технологий

ОТЧЕТ  
О ЛАБОРАТОРНОЙ РАБОТЕ № 4

по теме: Запросы на выборку данных к БД PostgreSQL.  
Представления в PostgreSQL.  
по дисциплине: Проектирование и реализация баз  
данных

Специальность:

09.03.03 Мобильные и сетевые технологии

Проверил:

Говорова М.М. \_\_\_\_\_

Дата: «13» мая 2021г.

Оценка \_\_\_\_\_

Выполнил:

студент  
группы К3241

Шутов Д. Э.

Санкт-Петербург 2021 г.

**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 №9 “Оптовая база”

## ВЫПОЛНЕНИЕ

### 1. Название БД «Wholesale base»

### 2. Схема инфологической модели данных БД

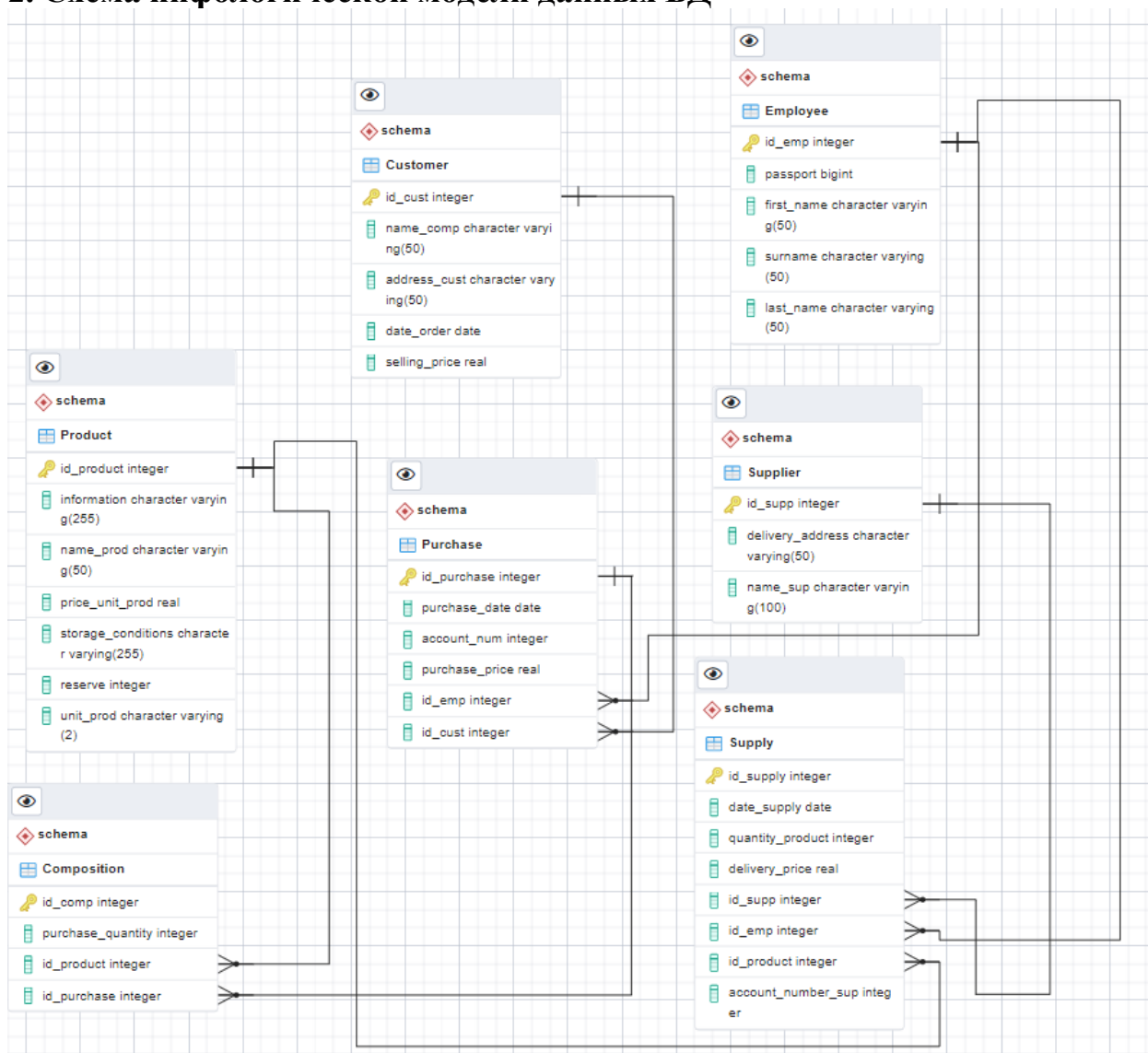


Рисунок 1 – Схема инфологической модели БД, сгенерированная в Generate ERD

### 3. Запросы к базе данных

- Вывести список поставщиков, которые поставляют все товары.

Wholesale base/postgres@PostgreSQL 11

Query Editor История запросов

```
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;
```

Результат План выполнения Сообщения Notifications

	name_sup	
	character varying (100)	
1	Аврора	

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

Wholesale base/postgres@PostgreSQL 11

Query Editor История запросов

```
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;
```

Результат План выполнения Сообщения Notifications

	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

```
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
```

Результат План выполнения Сообщения Notifications

	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	
1	<b>SELECT</b> SUM(purchase_price) <b>AS</b> result
2	<b>FROM</b> schema."Purchase"
3	<b>WHERE</b> purchase_date = <b>TIMESTAMP</b> 'yesterday';
Результат План выполнения Сообщения Notifications	
	<div> <div>result</div> <div>real</div> </div>
1	55465

- Вычислить стоимость товаров, оставшихся на складе

Wholesale base/postgres@PostgreSQL 11 ▾	
Query Editor История запросов	
1	<b>SELECT</b> name_prod, price_unit_prod*reserve <b>AS</b> result
2	<b>FROM</b> schema."Product"
3	<b>WHERE</b> reserve > 0;
Результат План выполнения Сообщения Notifications	
	<div> <div>name_prod</div> <div>character varying (50)</div> </div> <div> <div>result</div> <div>double precision</div> </div>
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	
1	<b>SELECT</b> date_supply, quantity_product
2	<b>FROM</b> schema."Supply"
3	<b>WHERE</b> quantity_product = ( <b>SELECT</b> min(quantity_product) <b>FROM</b> schema."Supply");
Результат План выполнения Сообщения Notifications	
	<div> <div>date_supply</div> <div>date</div> </div> <div> <div>quantity_product</div> <div>integer</div> </div>
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;
3
4

```

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

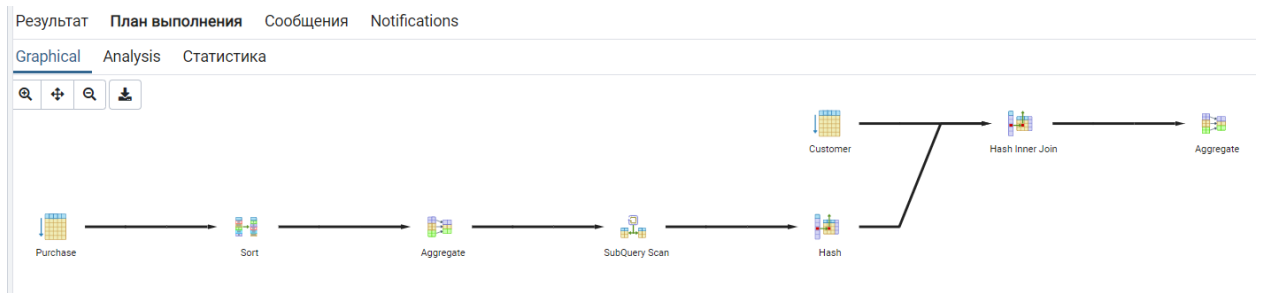
```

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 msec.

## 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 ▾

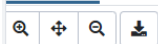
Query Editor История запросов

```

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 ценно 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 UPDATE schema."Purchase"
2 set purchase_price = 3000
3 where id_cust in
4     (select id_cust from schema."Customer"
5      where date_order < '2020-02-15');
```

Результат План выполнения Сообщения Notifications

UPDATE 2

Запрос завершён успешно, время выполнения: 131 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.

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 msec.

До:

Результат		План выполнения	Сообщения	Notifications	
	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		План выполнения		Сообщения		Notifications	
	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.