Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшегообразования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №4.2 «Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными»
«Анализ данных. Построение инфологической модели данных»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Дерещук Татьяна Евгеньевна

Факультет:ИКТ

Группа: К3140

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Цель работы

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

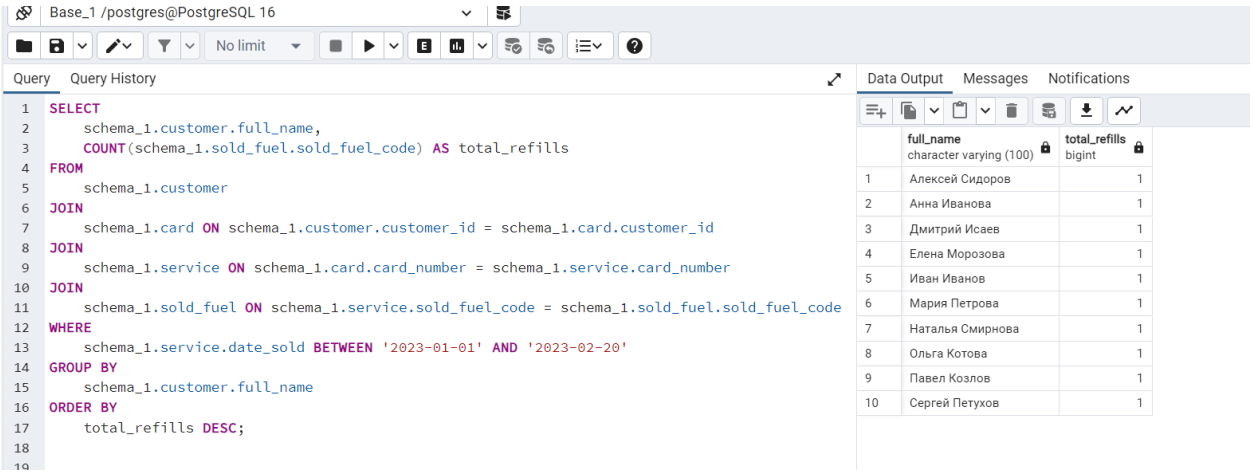
Практическое задание

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

Выполнение

Создать запросы:

- *Сколько раз заправлял автомобиль каждый из клиентов за заданный период.*

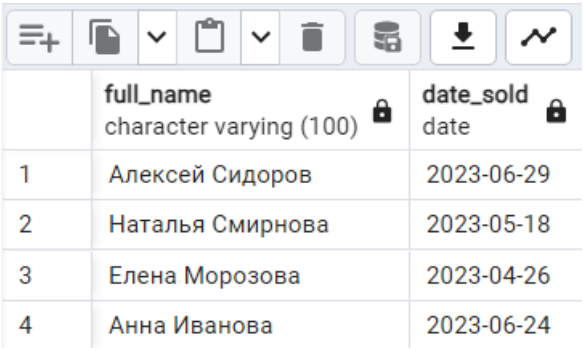



```
SELECT
    schema_1.customer.full_name,
    COUNT(schema_1.sold_fuel.sold_fuel_code) AS total_refills
FROM
    schema_1.customer
JOIN
    schema_1.card ON schema_1.customer.customer_id = schema_1.card.customer_id
JOIN
    schema_1.service ON schema_1.card.card_number = schema_1.service.card_number
JOIN
    schema_1.sold_fuel ON schema_1.service.sold_fuel_code = schema_1.sold_fuel.sold_fuel_code
WHERE
    schema_1.service.date_sold BETWEEN '2023-01-01' AND '2023-02-20'
GROUP BY
    schema_1.customer.full_name
ORDER BY
    total_refills DESC;
```

	full_name character varying (100)	total_refills bigint
1	Алексей Сидоров	1
2	Анна Иванова	1
3	Дмитрий Исаев	1
4	Елена Морозова	1
5	Иван Иванов	1
6	Мария Петрова	1
7	Наталья Смирнова	1
8	Ольга Котова	1
9	Павел Козлов	1
10	Сергей Петухов	1

- *Кто из клиентов не приобретал топливо в июле текущего года?*

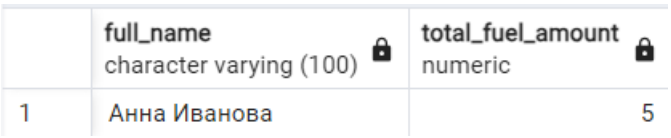
```
SELECT c.full_name, s.date_sold

FROM schema_1.customer c

LEFT JOIN schema_1.card cr ON c.customer_id = cr.customer_id

LEFT JOIN schema_1.service_1 s ON cr.card_number = s.card_number

WHERE s.date_sold IS NULL OR (EXTRACT(MONTH FROM s.date_sold) != 7 OR EXTRACT(YEAR FROM s.date_sold) != EXTRACT(YEAR FROM CURRENT_DATE))
```



	full_name character varying (100)	date_sold date
1	Алексей Сидоров	2023-06-29
2	Наталья Смирнова	2023-05-18
3	Елена Морозова	2023-04-26
4	Анна Иванова	2023-06-24

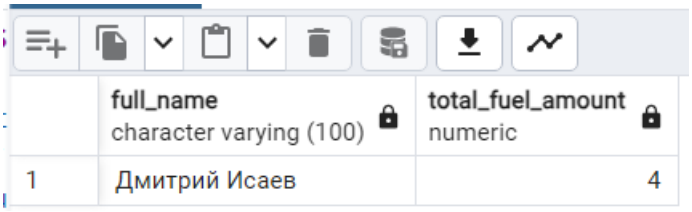
- *Найти клиента, купившего наибольший объем топлива по всей сети.*

```
SELECT c.full_name, SUM(s.amount_sold_fuel) as total_fuel_amount
FROM schema_1.customer c
JOIN schema_1.card cd ON c.customer_id = cd.customer_id
JOIN schema_1.service_1 s ON cd.card_number = s.card_number
GROUP BY c.full_name
ORDER BY total_fuel_amount DESC
```



	full_name character varying (100)	total_fuel_amount numeric
1	Анна Иванова	5

- *Вывести данные клиента, купившего топлива на наибольшую сумму в заданный день.*



	full_name character varying (100)	total_fuel_amount numeric
1	Дмитрий Исаев	4

```
SELECT customer.full_name, SUM(service.amount_sold_fuel) AS total_fuel_amount

FROM schema_1.customer customer

JOIN schema_1.card card ON customer.customer_id = card.customer_id

JOIN schema_1.service_1 service ON card.card_number = service.card_number

JOIN schema_1.fuel fuel ON service.sold_fuel_code = fuel.fuel_code

WHERE service.date_sold = '2023-07-27'

GROUP BY customer.full_name

ORDER BY total_fuel_amount DESC

LIMIT 1;
```

- *Какое топливо пользуется наибольшим спросом в прошедшем году на АЗС конкретного поставщика?*

```
SELECT
   schema_1.fuel.brand,
   schema_1.fuel.type,
   SUM(schema_1.service_1.amount_sold_fuel) AS total_amount
FROM
   schema_1.service_1
JOIN
   schema_1.sold_fuel ON schema_1.service_1.sold_fuel_code = schema_1.sold_fuel.sold_fuel_code
JOIN
   schema_1.fuel ON schema_1.sold_fuel.fuel_code = schema_1.fuel.fuel_code
JOIN
   schema_1.gas_station ON schema_1.sold_fuel.gas_station_code =
schema_1.gas_station.gas_station_code
JOIN
   schema_1.fuel_supplier_company ON schema_1.gas_station.fuel_type =
schema_1.fuel_supplier_company.fuel_type
WHERE
   schema_1.fuel_supplier_company.date_at_the_end >= CURRENT_DATE - INTERVAL '1 year'
GROUP BY
   schema_1.fuel.brand, schema_1.fuel.type
ORDER BY
   total_amount DESC
LIMIT 1;
```

	brand character varying (20)	type character varying (20)	total_amount numeric
1	Газ	Пропан	13

- *Какая из заправок продала топлива на наибольшую сумму по всем автозаправкам за последний год?*

Query | Query History

```
SELECT
    gs.gas_station_code,
    fsc.company_title AS supplier_company,
    SUM(s.amount_sold_fuel) AS total_fuel_sold
FROM
    schema_1.service_1 s
JOIN schema_1.sold_fuel sf ON s.sold_fuel_code
JOIN schema_1.gas_station gs ON sf.gas_station
JOIN schema_1.fuel_supplier_company fsc ON sf.
WHERE
    s.date_sold >= CURRENT_DATE - INTERVAL '1
GROUP BY
    gs.gas_station_code, fsc.company_title
ORDER BY
    total_fuel_sold DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	gas_station_code bigint	supplier_company character varying (100)	total_fuel_sold numeric
1	9	ОАО "Нефть-Газ-Продукт"	5

Создать представление:

- *Содержащее сведения обо всех АЗС и всех видах топлива, которые они продают;*

```
CREATE OR REPLACE VIEW schema_1.gas_station_fu
SELECT
    gs.gas_station_code,
    gs.company_phone,
    f.fuel_code,
    f.brand AS fuel_brand,
    f.type AS fuel_type
FROM
    schema_1.gas_station gs
JOIN schema_1.fuel_supplier_company fsc ON gs.
JOIN schema_1.fuel f ON fsc.fuel_type = f.fuel
```

```
CREATE VIEW

Query returned successfully in 80 msec.
```

Query | Query History

```
1  SELECT * FROM schema_1.gas_station_fuel_info;
```

Data Output | Messages | Notifications

	gas_station_code bigint	company_phone bigint	fuel_code bigint	fuel_brand character varying (20)	fuel_type character
1	1	9876543210	1	Бензин	АИ-95
2	2	1234567890	2	Дизель	DT-Л
3	3	5678901234	3	Газ	Пропан
4	4	8901234567	1	Бензин	АИ-95
5	5	9012345678	2	Дизель	DT-Л
6	6	3456789012	3	Газ	Пропан
7	7	9999999999	1	Бензин	АИ-95
8	8	8888888888	2	Дизель	DT-Л
9	9	7777777777	3	Газ	Пропан
10	10	6666666666	1	Бензин	АИ-95

- *Самая прибыльная АЗС за истекший месяц для каждого производителя.*

Query | Query History

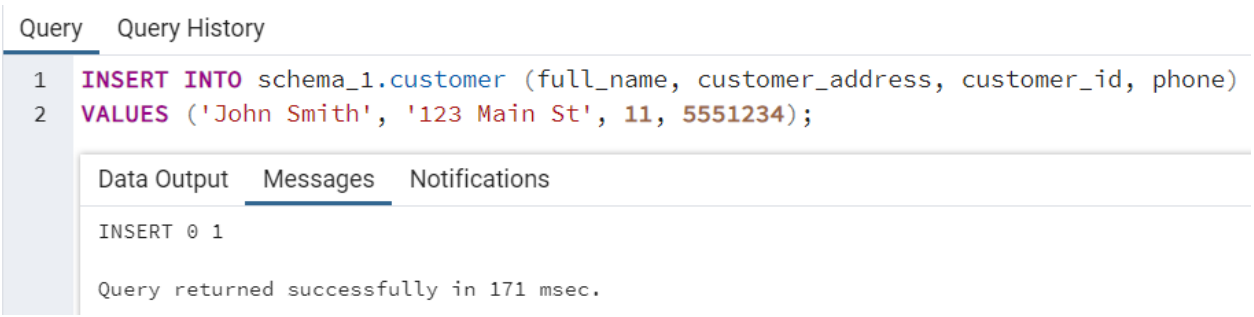
```
1  CREATE OR REPLACE VIEW schema_1.most_profitable_gas_stations AS
2  SELECT DISTINCT ON (f.brand, EXTRACT(MONTH FROM sf.price_end_date))
3  f.brand AS manufacturer,
4  gs.gas_station_code AS gas_station,
5  sf.price * s.amount_sold_fuel AS profit
6  FROM schema_1.fuel_supplier_company fsc
7  JOIN schema_1.gas_station gs ON fsc.fuel_type = gs.fuel_type
8  JOIN schema_1.sold_fuel sf ON gs.gas_station_code = sf.gas_station_code
9  JOIN schema_1.service_1 s ON sf.sold_fuel_code = s.sold_fuel_code
10 JOIN schema_1.fuel f ON fsc.fuel_type = f.fuel_code
11 WHERE EXTRACT(MONTH FROM sf.price_end_date) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '1 month')
12 ORDER BY f.brand, EXTRACT(MONTH FROM sf.price_end_date), profit DESC;
```

Data Output | Messages | Notifications

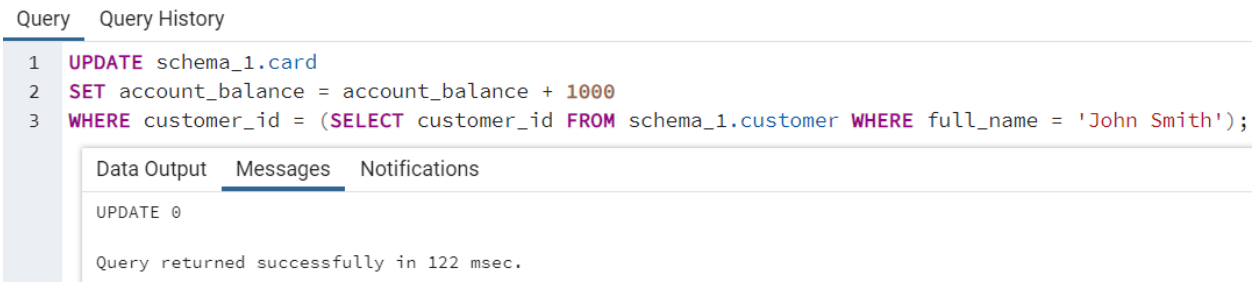
```
CREATE VIEW

Query returned successfully in 58 msec.
```

Составь 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

- Этот запрос вставляет новую запись в таблицу "customer" с указанными данными о клиенте.



- Этот запрос обновляет баланс карты клиента с именем "John Smith", увеличивая его на 1000.



Свои запросы:

- *Вывести всех клиентов с их адресами и номерами телефонов (без индексов)*

- *Вывести всех клиентов с их адресами и номерами телефонов (с индексами)*

Query Query History

```
1  SELECT c.full_name, c.customer_address, c.phone
2  FROM schema_1.customer c;
```

Data Output | Messages | Notifications

```
Successfully run. Total query runtime: 89 msec.
10 rows affected.
```

Простой: CREATE INDEX idx_customer_address ON schema_1.customer(customer_address);

Составной: CREATE INDEX idx_customer_address_phone ON schema_1.customer(customer_address, phone);

Data Output | Messages | Explain | Notifications

Graphical | Analysis | Statistics

customer

- *Получить информацию о всех проданных типах топлива, их брендах и количестве проданного топлива (без индексов)*

Query Query History

```
1  SELECT f.brand, f.type, sf.price, sf.price_start_date, sf.price_end_date, s1.amount_sold_fuel
2  FROM schema_1.sold_fuel sf
3  JOIN schema_1.fuel f ON sf.fuel_code = f.fuel_code
4  JOIN schema_1.service_1 s1 ON sf.sold_fuel_code = s1.sold_fuel_code;
```

Data Output | Messages | Notifications

```
Successfully run. Total query runtime: 110 msec.
10 rows affected.
```

- *Получить информацию о всех проданных типах топлива, их брендах и количестве проданного топлива (с индексами)*

Query | Query History

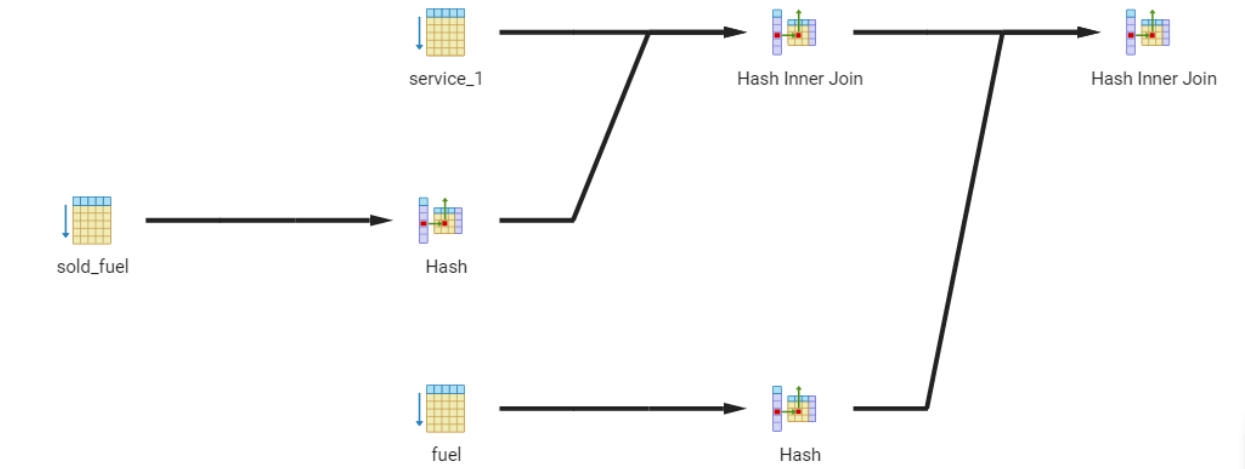
```
1  SELECT f.brand, f.type, s1.amount_sold_fuel
2  FROM schema_1.sold_fuel sf
3  JOIN schema_1.fuel f ON sf.fuel_code = f.fuel_code
4  JOIN schema_1.service_1 s1 ON sf.sold_fuel_code = s1.sold_fuel_code;
```

Data Output | Messages | Notifications

```
Successfully run. Total query runtime: 74 msec.
10 rows affected.
```

Data Output | Messages | Explain | Notifications

Graphical | Analysis | Statistics



Индексы:
Простой: CREATE INDEX idx_fuel_code ON schema_1.sold_fuel (fuel_code);
Составной: CREATE INDEX idx_brand_type ON schema_1.fuel (brand, type);