**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ**

**по Лабораторной работе № 4**

**«АНАЛИЗ ДАННЫХ. ПОСТРОЕНИЕ ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ БД»**

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

**Обучающийся** Проскуряков Роман Владимирович

**Факультет** прикладной информатики

**Группа** K3239

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Мобильные и сетевые технологии 2023

**Преподаватель** Говорова Марина Михайловна

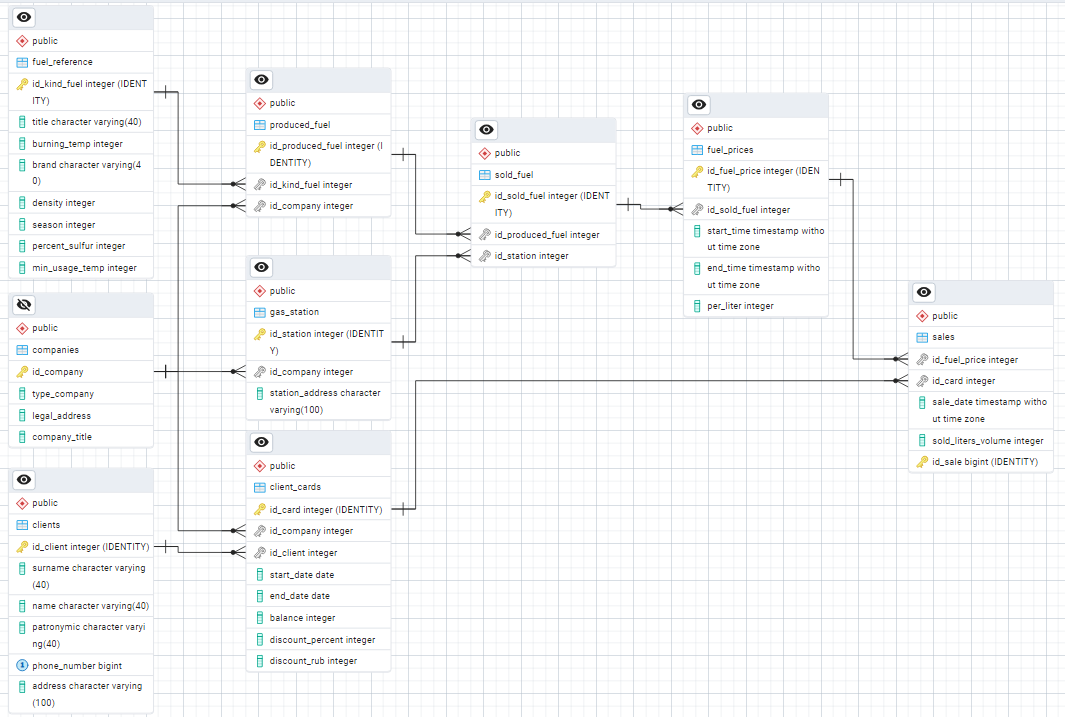
Санкт-Петербург

2024/2025

**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.

****

**Выполнение:**

1. Запросы и представления на выборку данных к базе данных PostgreSQL

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

SELECT clients.id\_client, clients.surname, clients.name, clients.patronymic, count(id\_sale) FROM

clients

LEFT JOIN client\_cards

ON clients.id\_client = client\_cards.id\_client

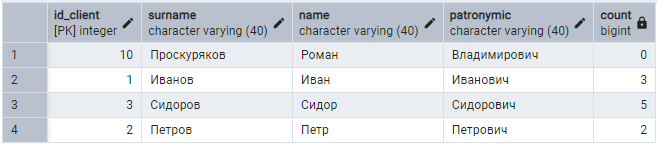
LEFT JOIN sales

ON client\_cards.id\_card = sales.id\_card

WHERE sale\_date is NULL OR

(sale\_date >= '2023-01-01 00:00:00'

AND sale\_date <= '2024-01-10 00:00:00')

GROUP BY clients.id\_client 

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

SELECT clients.id\_client, clients.surname, clients.name, clients.patronymic FROM

clients

LEFT JOIN client\_cards

ON clients.id\_client = client\_cards.id\_client

LEFT JOIN sales

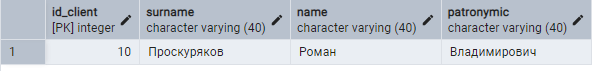
ON client\_cards.id\_card = sales.id\_card

WHERE sale\_date is NULL OR

(EXTRACT(YEAR FROM sale\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

AND EXTRACT(MONTH FROM sale\_date) = 7)

GROUP BY clients.id\_client

HAVING count(id\_sale) = 0 

* Найти клиента, купившего наибольший объем топлива по всей сети за истекший месяц.

WITH amount AS

(

SELECT clients.id\_client, clients.surname, clients.name, clients.patronymic, sum(sold\_liters\_volume) AS S FROM

clients

LEFT JOIN client\_cards

ON clients.id\_client = client\_cards.id\_client

LEFT JOIN sales

ON client\_cards.id\_card = sales.id\_card

WHERE sale\_date >= date\_trunc('month', CURRENT\_DATE) - interval '1 month'

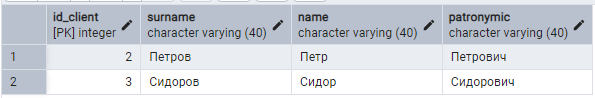
AND sale\_date < date\_trunc('month', CURRENT\_DATE)

GROUP BY clients.id\_client

)

SELECT amount.id\_client, amount.surname, amount.name, amount.patronymic FROM

amount

WHERE amount.S in (SELECT max(amount.S) FROM amount) 

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

WITH amount AS

(

SELECT clients.id\_client, clients.surname, clients.name, clients.patronymic,

sum((sold\_liters\_volume \* per\_liter) \* (100 - discount\_percent) / 100 - discount\_rub) AS S FROM

clients

LEFT JOIN client\_cards

ON clients.id\_client = client\_cards.id\_client

LEFT JOIN sales

ON client\_cards.id\_card = sales.id\_card

LEFT JOIN fuel\_prices

ON fuel\_prices.id\_fuel\_price = sales.id\_fuel\_price

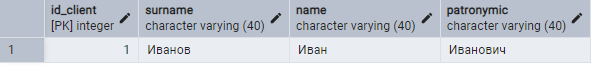
WHERE sale\_date::date = DATE('2025-07-04')

GROUP BY clients.id\_client

)

SELECT amount.id\_client, amount.surname, amount.name, amount.patronymic FROM

amount

WHERE amount.S in (SELECT max(amount.S) FROM amount)

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

WITH amount AS

(

SELECT produced\_fuel.id\_produced\_fuel, sum(sold\_liters\_volume) AS S FROM

produced\_fuel

LEFT JOIN fuel\_assortment

ON produced\_fuel.id\_produced\_fuel=fuel\_assortment.id\_produced\_fuel

LEFT JOIN filling\_stations

ON fuel\_assortment.id\_filling\_station=filling\_stations.id\_filling\_station

LEFT JOIN fuel\_prices

ON fuel\_assortment.id\_fuel\_offered=fuel\_prices.id\_fuel\_offered

LEFT JOIN sales

ON fuel\_prices.id\_fuel\_price=sales.id\_fuel\_price

WHERE sale\_date >= date\_trunc('year', CURRENT\_DATE) - interval '1 year'

AND sale\_date < date\_trunc('year', CURRENT\_DATE)

AND filling\_stations.id\_company\_owner=11

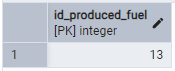
GROUP BY produced\_fuel.id\_produced\_fuel

)

SELECT amount.id\_produced\_fuel FROM

amount

WHERE amount.S in (SELECT max(amount.S) FROM amount)



* Сколько топлива каждого вида было продано за прошедший месяц по каждому поставщику на каждой заправке.

SELECT fuel\_reference.id\_kind\_fuel, id\_company\_factory, filling\_stations.id\_filling\_station,

COALESCE(sum(sold\_liters\_volume), 0) AS S FROM

fuel\_reference

LEFT JOIN

produced\_fuel

ON fuel\_reference.id\_kind\_fuel=produced\_fuel.id\_kind\_fuel

LEFT JOIN fuel\_assortment

ON produced\_fuel.id\_produced\_fuel=fuel\_assortment.id\_produced\_fuel

LEFT JOIN filling\_stations

ON fuel\_assortment.id\_filling\_station=filling\_stations.id\_filling\_station

LEFT JOIN fuel\_prices

ON fuel\_assortment.id\_fuel\_offered=fuel\_prices.id\_fuel\_offered

LEFT JOIN

(SELECT \* FROM sales

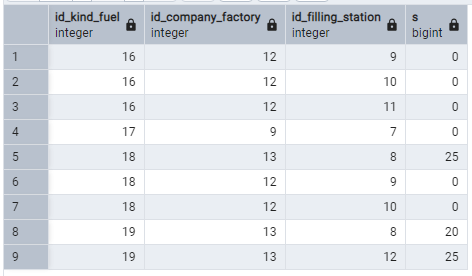
WHERE sale\_date >= date\_trunc('month', CURRENT\_DATE) - interval '1 month'

AND sale\_date < date\_trunc('month', CURRENT\_DATE)

) AS T

ON fuel\_prices.id\_fuel\_price=T.id\_fuel\_price

GROUP BY fuel\_reference.id\_kind\_fuel, filling\_stations.id\_filling\_station, id\_company\_factory

ORDER BY fuel\_reference.id\_kind\_fuel 

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

WITH amount AS

(

SELECT filling\_stations.id\_filling\_station,

COALESCE(sum((sold\_liters\_volume \* per\_liter) \* (100 - discount\_percent) / 100 - discount\_rub), 0) AS S

FROM

filling\_stations

LEFT JOIN fuel\_assortment

ON fuel\_assortment.id\_filling\_station=filling\_stations.id\_filling\_station

LEFT JOIN fuel\_prices

ON fuel\_assortment.id\_fuel\_offered=fuel\_prices.id\_fuel\_offered

LEFT JOIN

(SELECT \* FROM sales

WHERE EXTRACT(YEAR FROM sale\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

) AS T

ON fuel\_prices.id\_fuel\_price=T.id\_fuel\_price

LEFT JOIN client\_cards

ON T.id\_card=client\_cards.id\_card

GROUP BY filling\_stations.id\_filling\_station

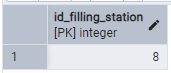
ORDER BY filling\_stations.id\_filling\_station

)

SELECT amount.id\_filling\_station FROM

amount

WHERE amount.S in (SELECT max(amount.S) FROM amount)



1. Представления

* Создать представление содержащее сведения обо всех АЗС и всех видах топлива, которые они продают

CREATE VIEW gas\_station\_fuel\_summary AS

SELECT filling\_stations.id\_filling\_station, id\_kind\_fuel FROM

filling\_stations

LEFT JOIN

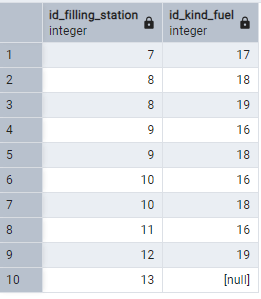
fuel\_assortment

ON fuel\_assortment.id\_filling\_station=filling\_stations.id\_filling\_station

LEFT JOIN

produced\_fuel

ON produced\_fuel.id\_produced\_fuel=fuel\_assortment.id\_produced\_fuel

ORDER BY filling\_stations.id\_filling\_station, id\_kind\_fuel

* Создать представление самая прибыльная АЗГС за истекший месяц для каждого производителя.

CREATE VIEW best\_filling\_stations\_for\_month AS

(

WITH amount AS

(

SELECT filling\_stations.id\_company\_owner, filling\_stations.id\_filling\_station,

COALESCE(sum((sold\_liters\_volume \* per\_liter) \* (100 - discount\_percent) / 100 - discount\_rub), 0) AS S

FROM

filling\_stations

LEFT JOIN fuel\_assortment

ON fuel\_assortment.id\_filling\_station=filling\_stations.id\_filling\_station

LEFT JOIN fuel\_prices

ON fuel\_assortment.id\_fuel\_offered=fuel\_prices.id\_fuel\_offered

LEFT JOIN

(SELECT \* FROM sales

WHERE sale\_date >= date\_trunc('month', CURRENT\_DATE) - interval '1 month'

AND sale\_date < date\_trunc('month', CURRENT\_DATE)

) AS T

ON fuel\_prices.id\_fuel\_price=T.id\_fuel\_price

LEFT JOIN client\_cards

ON T.id\_card=client\_ cards.id\_card

GROUP BY filling\_stations.id\_filling\_station

ORDER BY filling\_stations.id\_company\_owner, filling\_stations.id\_filling\_station

)

SELECT amount.id\_company\_owner, amount.id\_filling\_station, amount.S FROM

amount

JOIN

(

SELECT amount.id\_company\_owner, max(amount.S) AS M FROM

amount

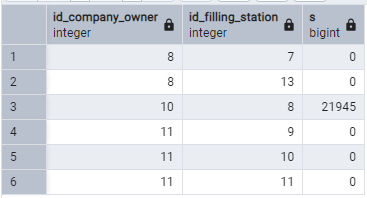
GROUP BY amount.id\_company\_owner

) AS T

ON amount.id\_company\_owner=T.id\_company\_owner

AND amount.S=T.M

)



1. 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) **с использованием подзапросов**

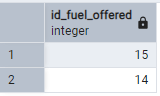
* INSERT

Были записи о наличии в ассортименте без указанной цены. Исправим это.

SELECT id\_fuel\_offered FROM fuel\_assortment

EXCEPT

SELECT id\_fuel\_offered FROM fuel\_prices



WITH T AS

(

SELECT id\_fuel\_offered FROM fuel\_assortment

EXCEPT

SELECT id\_fuel\_offered FROM fuel\_prices

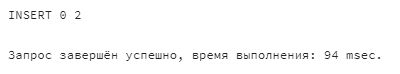
)

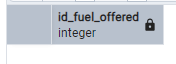
INSERT INTO fuel\_prices(id\_fuel\_offered, start\_time, end\_time, per\_liter)

(

SELECT id\_fuel\_offered, CURRENT\_DATE as start\_time, NULL as end\_time, 9999999 as per\_liter FROM T

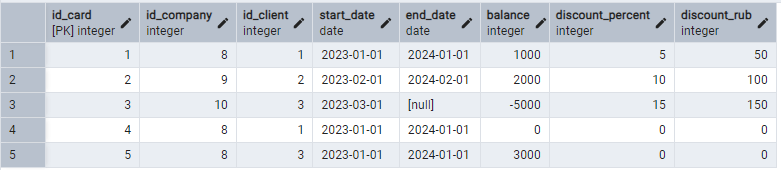
)





* UPDATE

Сделаем карты бессрочными, через которые, которые потратили у нас в сумме за предыдущий год больше, чем на 10000



UPDATE client\_cards SET end\_date=NULL WHERE id\_card IN

(

SELECT T2.id\_card FROM

(

SELECT client\_cards.id\_card,

COALESCE(sum((sold\_liters\_volume \* per\_liter) \* (100 - discount\_percent) / 100 - discount\_rub), 0) AS S

FROM

fuel\_prices

JOIN

(SELECT \* FROM sales

WHERE sale\_date >= date\_trunc('year', CURRENT\_DATE) - interval '1 year'

AND sale\_date < date\_trunc('year', CURRENT\_DATE)

) AS T

ON fuel\_prices.id\_fuel\_price=T.id\_fuel\_price

JOIN client\_cards

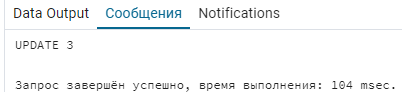
ON T.id\_card=client\_cards.id\_card

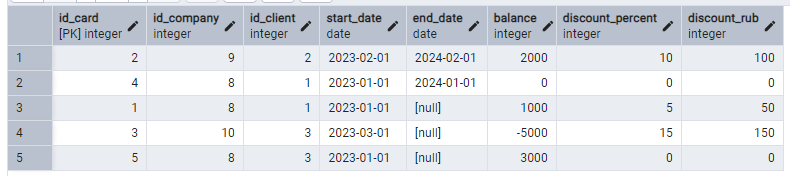
GROUP BY client\_cards.id\_card

ORDER BY client\_cards.id\_card

) as T2 WHERE T2.S > 10000

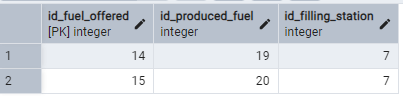
)





* DELETE

Удалим из ассортимента все топлива, которые ни разу не были проданы (мы добавили их в шаге INSERT)



DELETE FROM fuel\_assortment

WHERE NOT EXISTS

(

SELECT 1 FROM

fuel\_prices

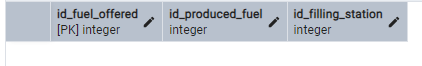
LEFT JOIN sales

ON sales.id\_fuel\_price=fuel\_prices.id\_fuel\_price

WHERE id\_sale is NOT NULL

AND fuel\_prices.id\_fuel\_offered = fuel\_assortment.id\_fuel\_offered

)



Также каскадно удалились и цены, т.к. я так настроил.

1. Простой и составной индексы для двух произвольных запросов и сравнение времени выполнения запросов без индексов и с индексами. Для получения плана запроса используется команда EXPLAIN

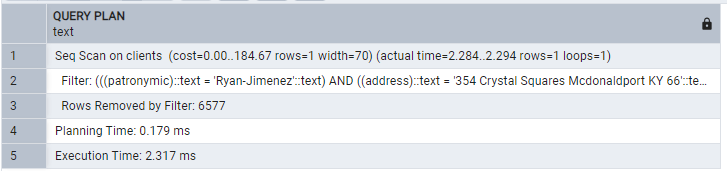
* Без индексов

EXPLAIN ANALYZE

SELECT \* FROM clients

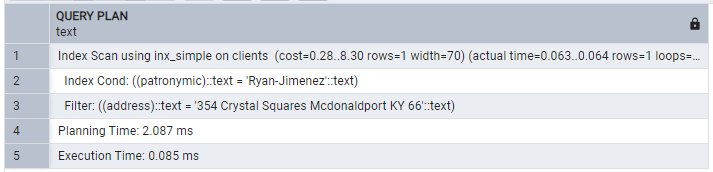
WHERE patronymic = 'Ryan-Jimenez'

AND address='354 Crystal Squares Mcdonaldport KY 66'



* С простым индексом

CREATE INDEX inx\_simple ON clients(patronymic)



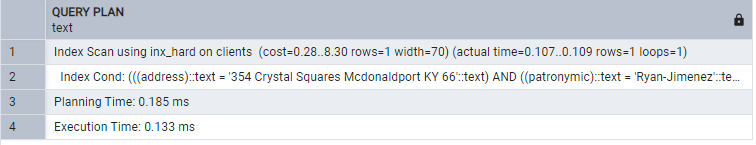
* С составным индексом

Сначала удалим прошлый идекс

DROP INDEX inx\_simple

Теперь создадим новый

CREATE INDEX inx\_hard ON clients(address, patronymic)



**С индексами с десятки раз быстрее!**

**Выводы**

В ходе выполнения лабораторной работы были освоены ключевые навыки работы с PostgreSQL, такие как создание запросов, модификацию данных, использование подзапросов, а также оптимизацию запросов через индексы. Полученные знания и опыт будут полезны для дальнейшей работы с базами данных и разработки эффективных баз данных.