**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

# Отчет

по лабораторной работе «ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,

ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL»по дисциплине «**Проектирование** **Баз данных**»

Автор:Зайцев Кирилл Дмитриевич

Факультет: Инфокоммуникационных технологий (ИКТ)

Группа: К32402

Преподаватель: Говорова М. М.

Дата: 14.05.2023



Санкт-Петербург 2023

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

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

1. **- 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.

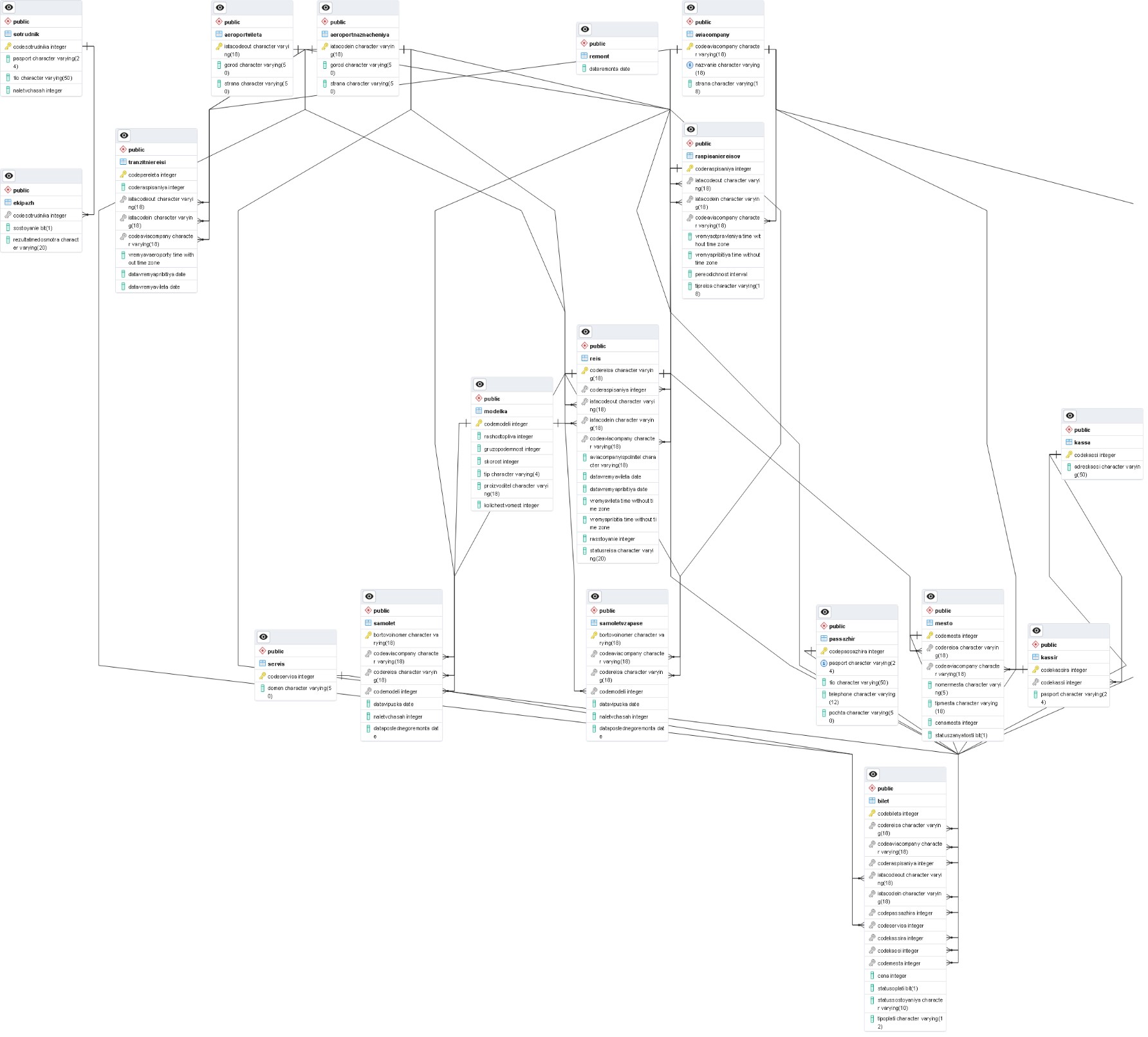
**Задание 2.** Создать запросы:

* + Определить расчетное время полета по всем маршрутам.
  + Определить расход топлива по всем маршрутам.
  + Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по заданному из рейсов за вчерашний день.
  + Рассчитать убытки компании за счет непроданных билетов за вчерашний день.
  + Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения.
  + Вывести список самолетов, “возраст” которых превышает средний “возраст” самолетов этого типа. • Определить тип самолетов, летающих во все аэропорты назначения.

**Задание 3.** Создать представление:

* + для пассажиров авиакомпании о рейсах в Москву на ближайшую неделю; • количество самолетов каждого типа, летавшими за последний месяц.

1. - Схема базы данных



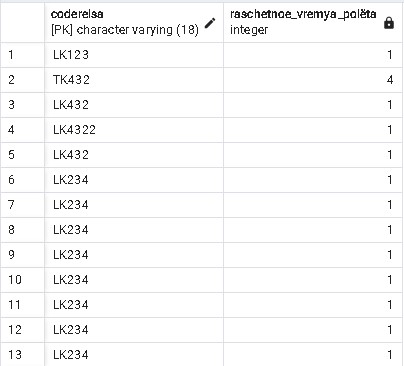
1. – Запросы:

* + Определить расчетное время полета по всем маршрутам.

Код:

SELECT r.codereisa, r.rasstoyanie / m.skorost AS raschetnoe\_vremya\_polёta FROM reis r JOIN samolet s ON s.codereisa = r.codereisa JOIN modelka m ON s.codeModeli = m.codeModeli WHERE rasstoyanie IS NOT NULL AND skorost IS NOT NULL;

Скриншот выполнения:



* + Определить расход топлива по всем маршрутам.

Код:

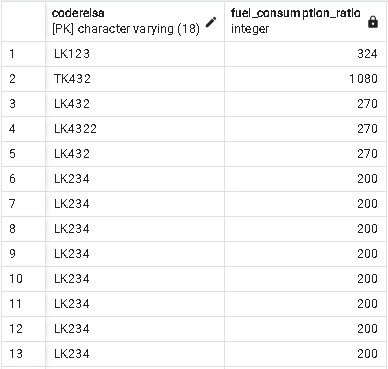
SELECT r.codereisa, r.rasstoyanie / 100 \* m.rashodTopliva AS fuel\_consumption\_ratio

FROM reis r

JOIN samolet s ON r.codereisa = s.codereisa

JOIN modelka m ON s.codemodeli = m.codemodeli;

Скриншот выполнения:



* + Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по заданному из рейсов за вчерашний день.

Код:

SELECT m.kolichestvomest - COUNT(b.codeBileta) AS svobodnye\_mesta

FROM modelka m

JOIN samolet s ON m.codeModeli = s.codeModeli

JOIN reis r ON s.bortovoiNomer = s.bortovoiNomer

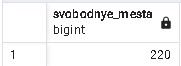
JOIN bilet b ON r.codereisa = b.codereisa AND s.bortovoiNomer = s.bortovoiNomer AND r.dataVremyaVileta = r.dataVremyaVileta

WHERE r.codereisa = 'LK234'

AND r.dataVremyaVileta = CURRENT\_DATE - INTERVAL '1 day'

GROUP BY m.codeModeli, m.kolichestvomest;

Скриншот выполнения:



* Рассчитать убытки компании за счет непроданных билетов за вчерашний день.

Код:

SELECT

SUM(b.cena) AS ubytki

FROM

bilet AS b

INNER JOIN

reis AS r ON b.codeReisa = r.codeReisa

WHERE

r.dataVremyaVileta = CURRENT\_DATE - INTERVAL '1 day'

AND b.statusSostoyaniya = 'cancelled';

ИЛИ

SELECT

SUM(m.cenaMesta) AS ubytki

FROM

mesto AS m

LEFT JOIN

bilet AS b ON m.codeMesta = b.codeMesta

LEFT JOIN

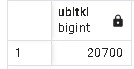
reis AS r ON b.codeReisa = r.codeReisa

WHERE

r.dataVremyaVileta = CURRENT\_DATE - INTERVAL '1 day'

AND m.statuszanyatisti = ‘0’;

Скриншот выполнения (второго варианта):



* Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения.

Код:

SELECT s.codeModeli, COUNT(\*) AS flights\_count

FROM samolet s

JOIN reis r ON s.codeReisa = r.codeReisa

WHERE r.IATACodeIN = 'FRFT'

GROUP BY s.codeModeli

ORDER BY flights\_count DESC

LIMIT 1;

Скриншот выполнения:



* Вывести список самолетов, “возраст” которых превышает средний “возраст” самолетов этого типа.

Код:

SELECT s.bortovoiNomer, s.codeModeli, s.dataVipuska, ((CURRENT\_DATE) - s.dataVipuska) AS age

FROM samolet s

JOIN (

SELECT codeModeli, AVG((CURRENT\_DATE) - dataVipuska) AS avg\_age

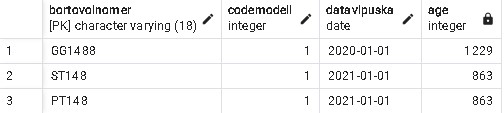
FROM samolet

GROUP BY codeModeli

) AS t ON s.codeModeli = t.codeModeli

WHERE (CURRENT\_DATE) - s.dataVipuska > t.avg\_age;

Скриншот выполнения:



• Определить тип самолетов, летающих во все аэропорты назначения.

Код:

SELECT s.codeModeli

FROM samolet s

JOIN reis r ON s.codereisa = r.codereisa

GROUP BY s.codeModeli

HAVING COUNT(DISTINCT r.IATACodeIN) = (SELECT COUNT(\*) FROM aeroportNaznacheniya)

Скриншот выполнения:



Создать представление:

-для пассажиров авиакомпании о рейсах в Франкфурт на ближайшую неделю

Код:

CREATE VIEW flights\_to\_frankfurt\_next\_week AS

SELECT r.codereisa, s.bortovoiNomer , r.dataVremyaVileta , r.VremyaVileta , r.dataVremyaPribitiya , r.VremyaPribitia

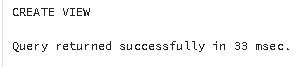
FROM samolet s

JOIN reis r ON s.codereisa = r.codereisa

WHERE r.IATACodeIN = 'FRFT'

AND r.dataVremyaVileta BETWEEN CURRENT\_DATE AND CURRENT\_DATE + INTERVAL '7 days' AND s.codemodeli = 2;

Скриншоты выполнения:





-количество самолетов каждого типа, летавшими за последний месяц

Код:

CREATE VIEW num\_flights\_by\_type AS

SELECT s.codeModeli, COUNT(DISTINCT r.codereisa) AS num\_flights

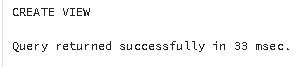
FROM samolet s

JOIN reis r ON s.codereisa = r.codereisa

WHERE r.datavremyavileta >= CURRENT\_DATE - INTERVAL '1 month'

GROUP BY s.codeModeli;

Скриншоты выполнения:



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

INSERT:

Добавление в таблицу запасных самолетов, тех самолетов, у которых давно не производился техосмотр.

Код:

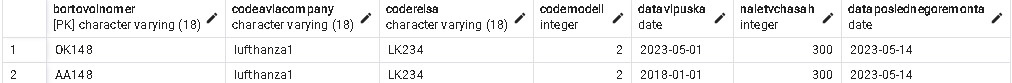
INSERT INTO samoletvzapase (bortovoiNomer , codeAviacompany , codeReisa , codeModeli , dataVipuska , naletVchasah , dataPoslednegoRemonta)

SELECT bortovoiNomer , codeAviacompany , codeReisa , codeModeli ,

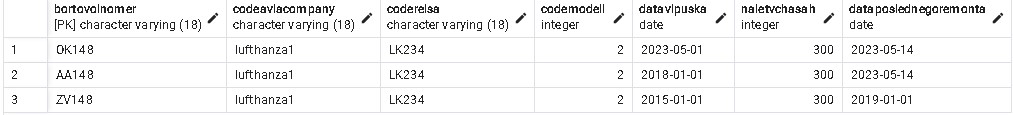
dataVipuska , naletVchasah , dataPoslednegoRemonta FROM samolet

WHERE dataPoslednegoRemonta < '01.01.2023';

Скриншот таблицы до команды:



Скриншот после выполнения запроса:



DELETE: Удаляет из списка активных самолётов те, которые были отправлены на ремонт/Т.О.

Код:

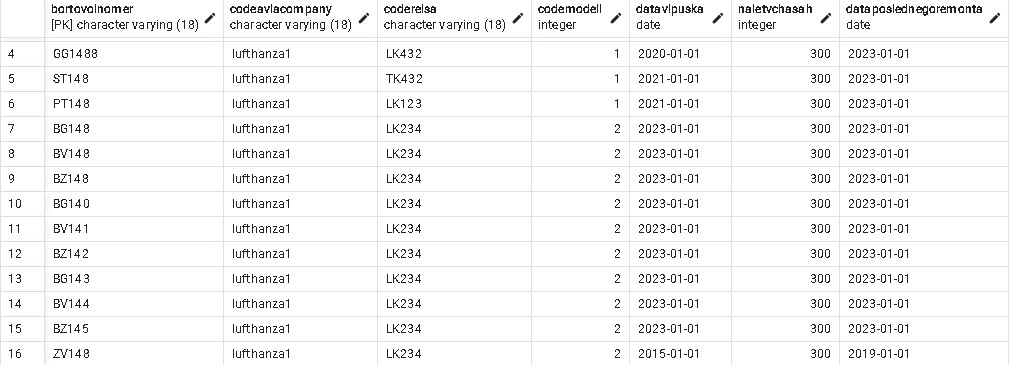
DELETE FROM samolet

WHERE bortovoiNomer IN

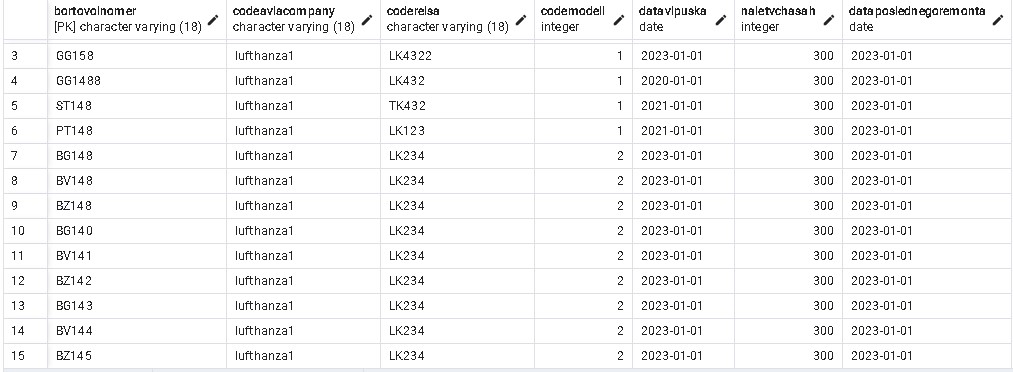
(SELECT bortovoiNomer FROM samoletvzapase)

AND dataPoslednegoRemonta < '2023-01-01';

Скриншот до выполнения:



Скриншот после выполнения:



UPDATE:

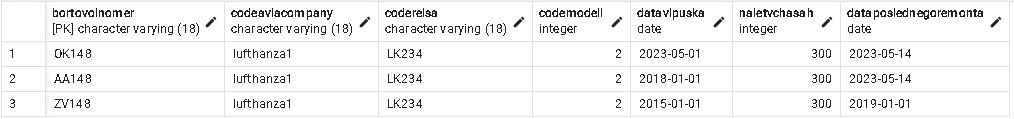
Обновляет данные о последнем ремонте/Т.О самолета находящегося на ремонте/Т.О.

Код:

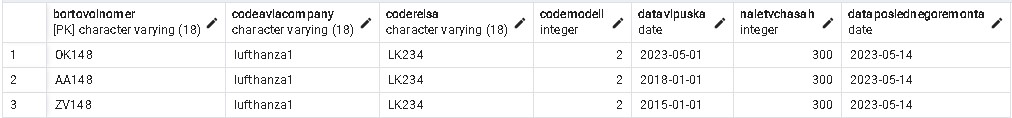
UPDATE samoletvzapase

SET dataPoslednegoRemonta = (SELECT dateremonta FROM remont)

Скриншот до:

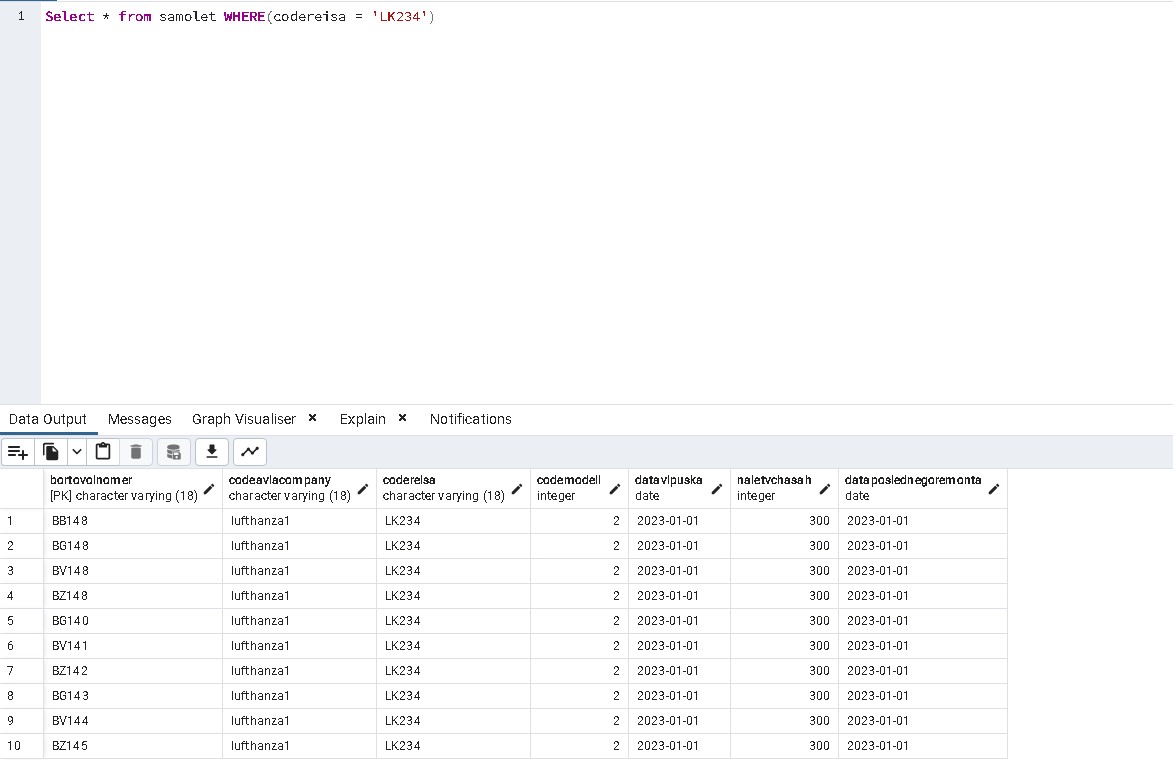


Скриншот после:

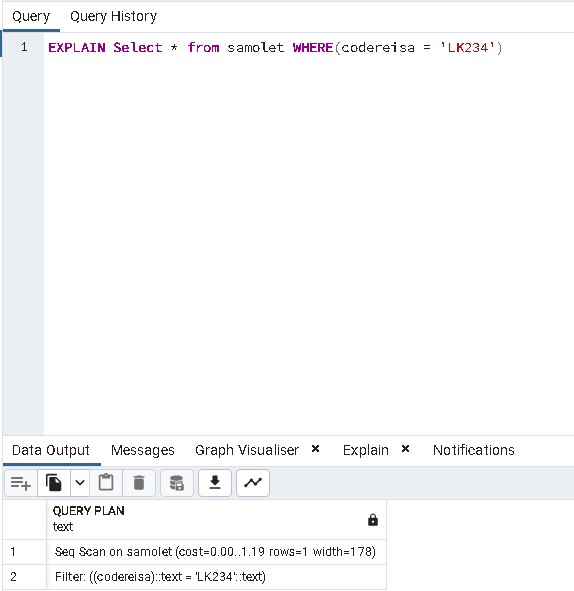


ИНДЕКСЫ:

Запрос без индекса:

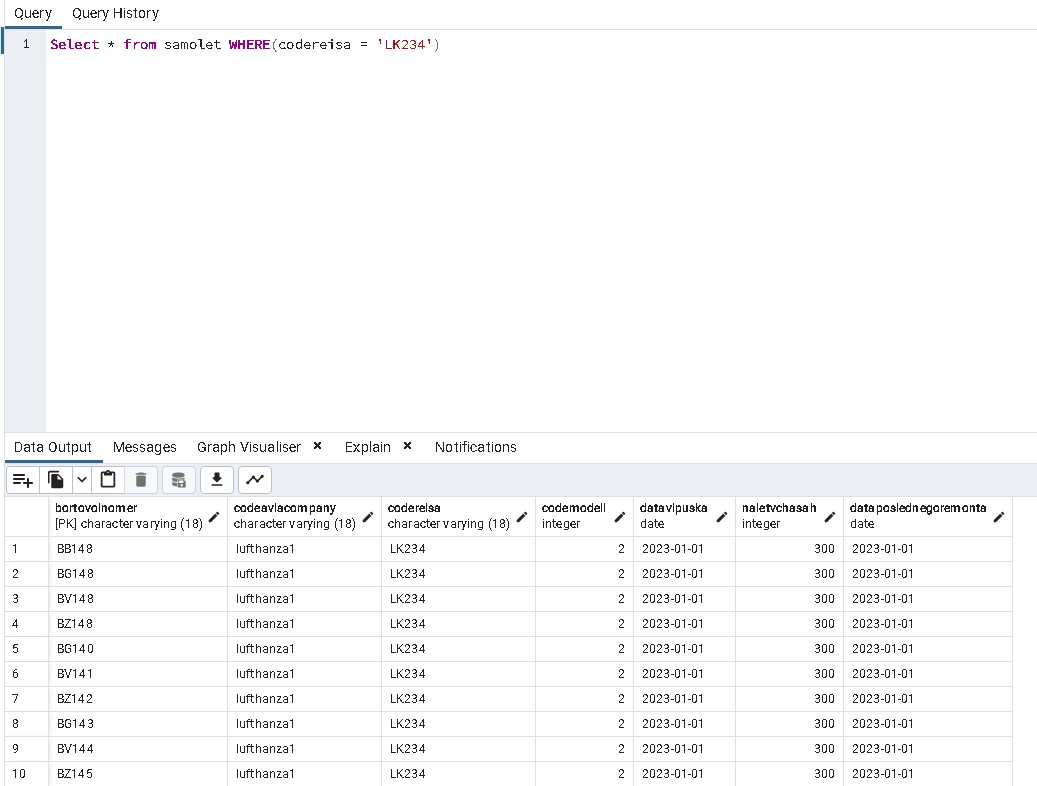


План запроса:

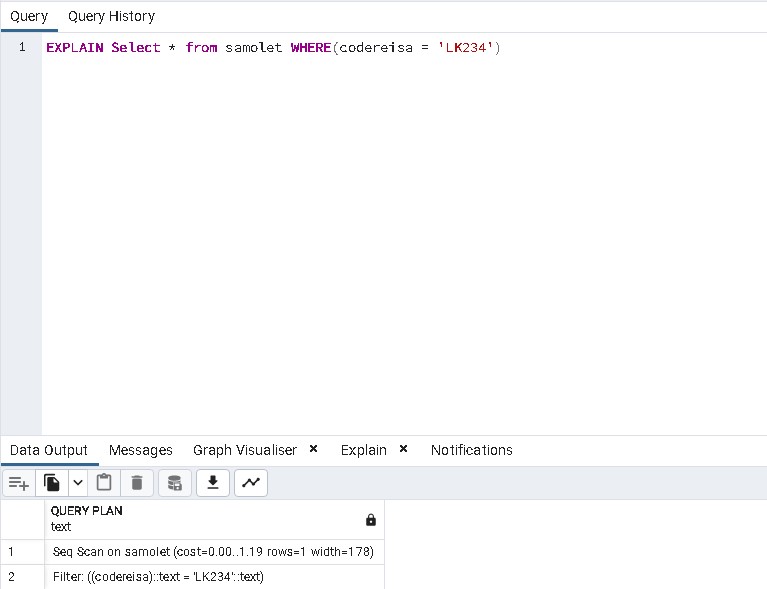


Код индекса:

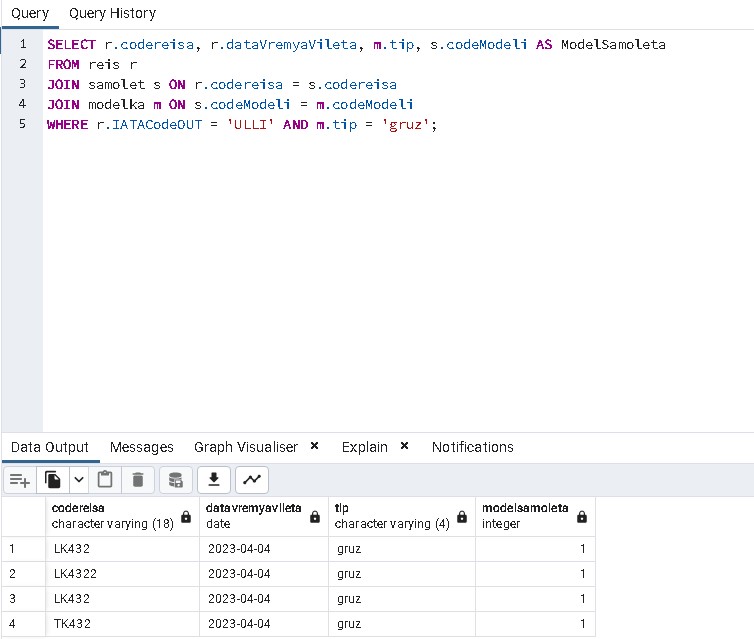
Create index idx\_codereisa on samolet (codereisa) Скриншот выполнения запроса:



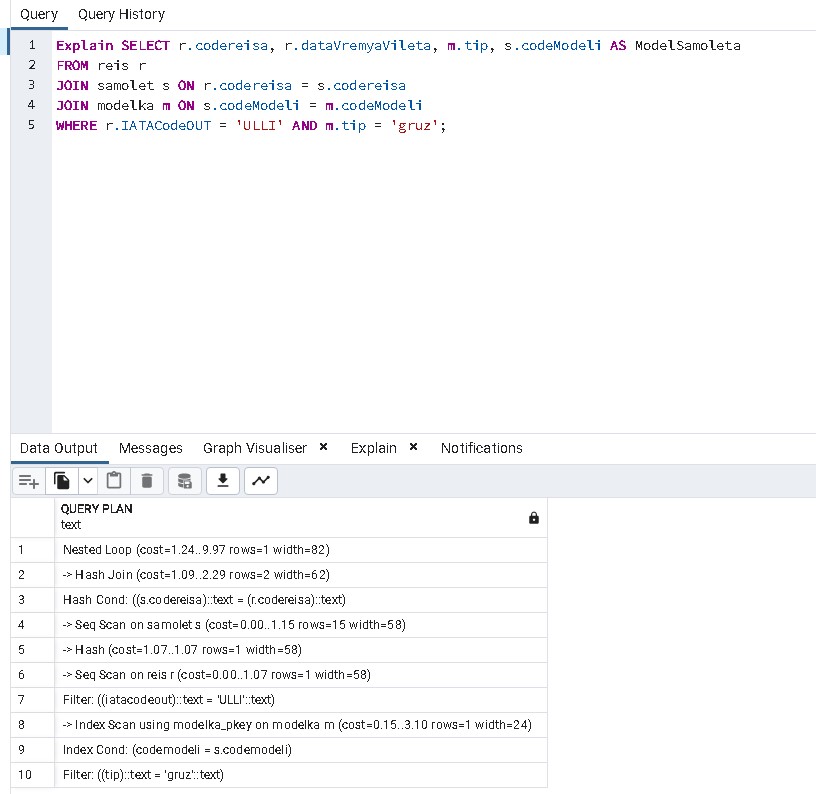
План запроса:



Запрос без индекса:



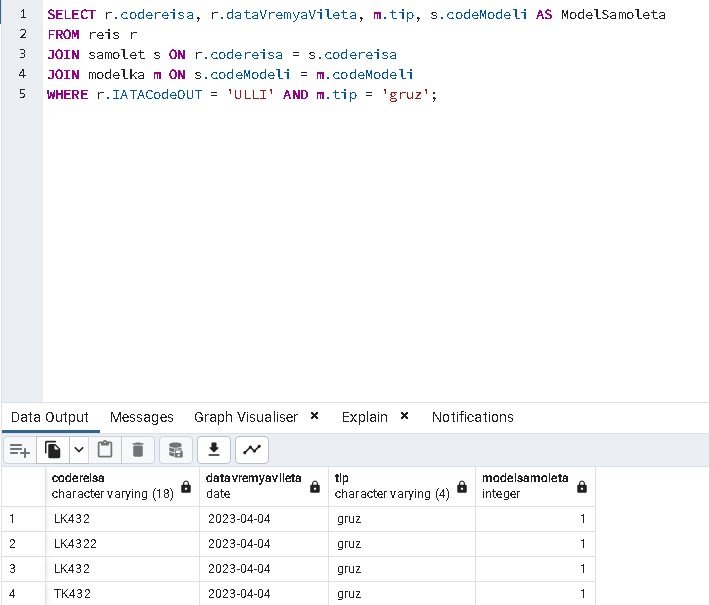
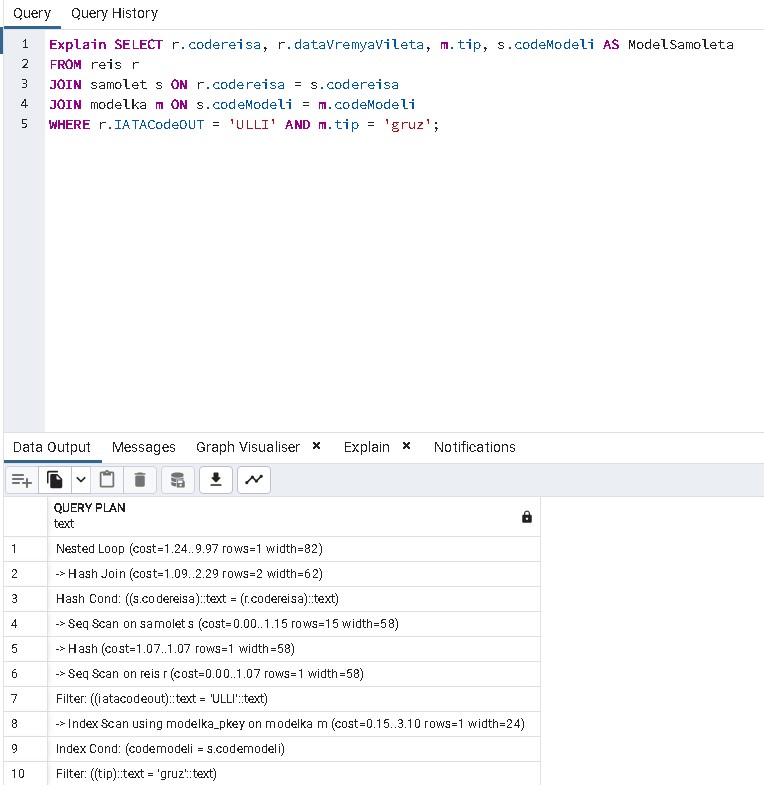
План запроса:



Создание индекса:

Create index idx\_codereisa\_codeModeli on samolet (codereisa, codeModeli)

Запрос с индексом:



План запроса:

После выполнения запросов:

-Как в первом, так и во втором случае выполнение запроса с индексом заняло больше времени, чем без него. Вероятнее всего по той причине, что таблица заполнена НЕ большим объёмом данных.

**Выводы:**

В результате выполнения данной работы я овладел практическими навыками работы с базой данных PostgreSQL. Я научился создавать представления и делать запросы на выборку данных, а также использовать подзапросы при модификации данных. Кроме того, я познакомился с индексами и научился создавать и использовать простые и составные индексы для ускорения выполнения запросов. В целом, выполнение данной работы помогло мне лучше понять, как работать с PostgreSQL и как оптимизировать запросы для более быстрого доступа к данным. Эти навыки могут быть полезными при разработке приложений, которые используют базы данных PostgreSQL.

**Список использованных источников:**

1. Документация PostgreSQL. Запросы [Электронный ресурс] // Официальный сайт PostgreSQL. 1996-2023. URL: <https://www.postgresql.org/docs/10/queries.html>(дата обращения: 02.03.2023).
2. Документация pgAdmin 4 PostgreSQL. Query Tool [Электронный ресурс] //

Официальный сайт pgAdmin . URL: <https://www.pgadmin.org/docs/pgadmin4/latest/query_tool.html>(дата обращения: 02.03.2023).