Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Базы данных

Отчёт по лабораторной работе №4

Вариант 5531

Выполнила:

*Волкова Ирина Александровна*

*P3115*

Проверил:

*Райла Мартин*

Санкт-Петербург 2024

# Задание

Составить запросы на языке SQL (пункты 1-2).

Для каждого запроса предложить индексы, добавление которых уменьшит время выполнения запроса (указать таблицы/атрибуты, для которых нужно добавить индексы, написать тип индекса; объяснить, почему добавление индекса будет полезным для данного запроса).

Для запросов 1-2 необходимо составить возможные планы выполнения запросов. Планы составляются на основании предположения, что в таблицах отсутствуют индексы. Из составленных планов необходимо выбрать оптимальный и объяснить свой выбор.  
Изменятся ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА.  
   Фильтры (AND):  
   a) Н\_ОЦЕНКИ.КОД < 3.  
   b) Н\_ВЕДОМОСТИ.ИД > 1490007.  
   c) Н\_ВЕДОМОСТИ.ИД < 39921.  
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ФАМИЛИЯ = Петров.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409.  
   Вид соединения: LEFT JOIN.

# 1.Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

a) Н\_ОЦЕНКИ.КОД < 3.

b) Н\_ВЕДОМОСТИ.ИД < 1490007.

c) Н\_ВЕДОМОСТИ.ИД > 39921.

Вид соединения: RIGHT JOIN.

## ЗАПРОС 1:

SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА

FROM Н\_ВЕДОМОСТИ

RIGHT JOIN Н\_ОЦЕНКИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД

WHERE Н\_ОЦЕНКИ.КОД = '2' AND Н\_ВЕДОМОСТИ.ИД < 1490007 AND Н\_ВЕДОМОСТИ.ИД > 39921;

### УМЕНЬШЕНИЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ 1 ЗАПРОСА

1) На таблице Н\_ОЦЕНКИ

* индекс на атрибуте КОД (B-tree). Это ускорит фильтрацию строк с условием Н\_ОЦЕНКИ.КОД = '2'

2) На таблице Н\_ВЕДОМОСТИ

* индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием Н\_ВЕДОМОСТИ.ИД < 1490007 и Н\_ВЕДОМОСТИ.ИД > 39921
* индекс на атрибуте КОД (B-tree). Это ускорит соединение таблиц по атрибуту КОД.

### ВОЗМОЖНЫЙ ПЛАН ВЫПОЛНЕНИЯ ЗАПРОСА БЕЗ ИНДЕКСОВ:

**План 1**

* Полный скан таблицы Н\_ВЕДОМОСТИ
* Полный скан таблицы Н\_ОЦЕНКИ с применением фильтра по условию Н\_ОЦЕНКИ.КОД = '2'
* Создание таблиц с использованием Nested Loop Join по атрибуту КОД
* Фильтрация результата по условию Н\_ВЕДОМОСТИ.ИД < 1490007, фильтрация результата по условию Н\_ВЕДОМОСТИ.ИД > 39921

**План 2**

* Полный скан таблицы Н\_ОЦЕНКИ с применением фильтра по условию Н\_ОЦЕНКИ.КОД = '2'
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию Н\_ВЕДОМОСТИ.ИД < 1490007, применение фильтра по условию Н\_ВЕДОМОСТИ.ИД > 39921
* Соединение таблиц с использованием Nested Loop Join по атрибуту КОД

**План 3**

* Сортировка атрибута КОД в таблице Н\_ОЦЕНКИ в естественном порядке
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию Н\_ВЕДОМОСТИ.ИД < 1490007, применение фильтра по условию Н\_ВЕДОМОСТИ.ИД > 39921
* Полный скан таблицы Н\_ОЦЕНКИ с применением фильтра по условию Н\_ОЦЕНКИ.КОД = '2'
* Соединение таблиц с использованием Merge Join по атрибуту КОД

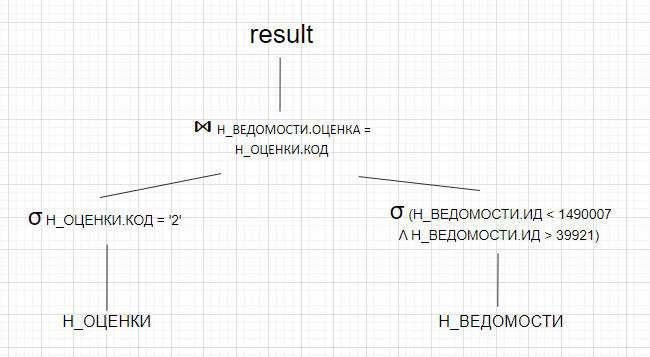
**План 4**

* Полный скан таблицы Н\_ОЦЕНКИ с применением фильтра по условию Н\_ОЦЕНКИ.КОД = '2'
* Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию Н\_ВЕДОМОСТИ.ИД < 1490007, применение фильтра по условию Н\_ВЕДОМОСТИ.ИД > 39921
* Соединение таблиц с использованием Hash Join по атрибуту КОД

### ОПТИМАЛЬНЫЙ ПЛАН:

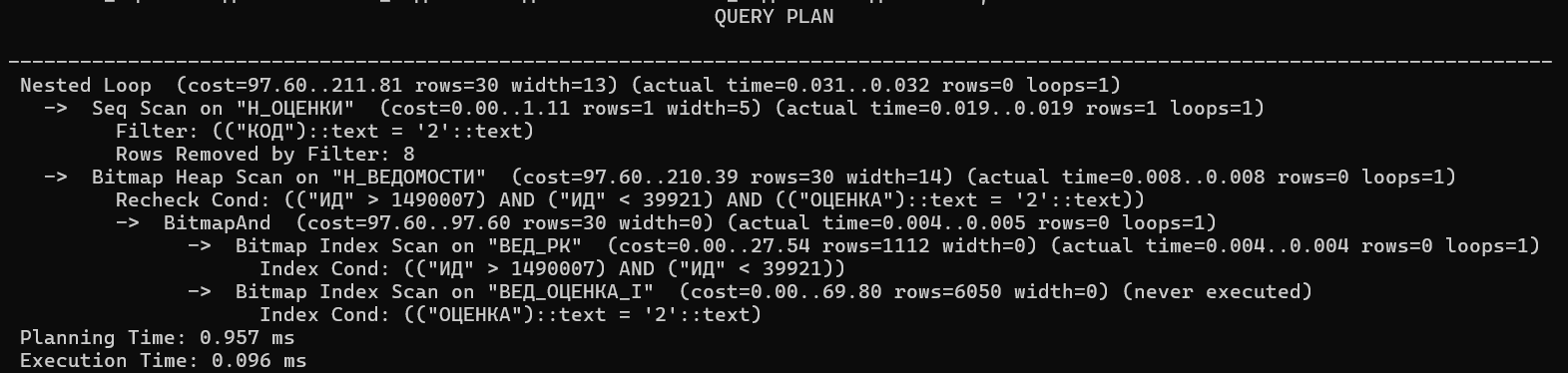
План 4, так как фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки. В данном случае предпочтительнее использовать Hash Join так как Nested Loop Join для RIGHT JOIN не используется (объединение, использующее RIGHT JOIN, которое будет работать с Nested Loop, внутренне трансформируется в LEFT JOIN, чтобы операция Nested Loop могла сработать). Для Merge Join понадобится сортировка, что потребует дополнительных усилий.

Не существует Nested Loop Right Join, потому что Nested Loop всегда начинается слева и берет левую сторону как основу для цикла. Поэтому объединение, использующее RIGHT JOIN, которое будет работать с Nested Loop, внутренне трансформируется в LEFT JOIN, чтобы операция Nested Loop могла сработать.



### ПРИ ДОБАВЛЕНИИ ИНДЕКСОВ ПЛАН ВЫПОЛНЕНИЯ ЗАПРОСА ИЗМЕНИТСЯ:

* Вместо полного скана таблиц будет использоваться индексный скан
* Hash Join станет быстрее благодаря индексу на атрибуте КОД



EXPLAIN ANALYZE SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ДАТА

FROM Н\_ВЕДОМОСТИ

RIGHT JOIN Н\_ОЦЕНКИ ON Н\_ВЕДОМОСТИ.ОЦЕНКА = Н\_ОЦЕНКИ.КОД

WHERE Н\_ОЦЕНКИ.КОД = '2' AND Н\_ВЕДОМОСТИ.ИД > 1490007 AND Н\_ВЕДОМОСТИ.ИД < 39921;

# 2.Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.

Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

a) Н\_ЛЮДИ.ФАМИЛИЯ = Петров.

b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409.

Вид соединения: LEFT JOIN.

## ЗАПРОС 2:

SELECT Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО

FROM Н\_ЛЮДИ

LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

LEFT JOIN Н\_УЧЕНИКИ ON Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ФАМИЛИЯ = 'Петров' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409;

### УМЕНЬШЕНИЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ 2 ЗАПРОСА

1) На таблице Н\_ЛЮДИ

* Индекс на атрибуте ФАМИЛИЯ (B-tree). Это ускорит фильтрацию строк с условием Н\_ЛЮДИ.ФАМИЛИЯ = 'Петров'

2) На таблице Н\_ОБУЧЕНИЯ

* Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409 и соединение таблиц по атрибуту ЧЛВК\_ИД

### ВОЗМОЖНЫЙ ПЛАН ВЫПОЛНЕНИЯ ЗАПРОСА БЕЗ ИНДЕКСОВ:

**План 1**

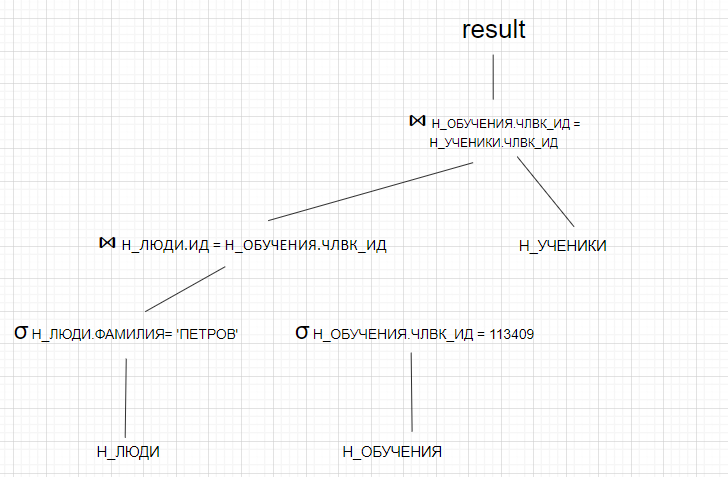
* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию Н\_ЛЮДИ.ФАМИЛИЯ = 'Петров'
* Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409
* Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД
* Полный скан таблицы Н\_УЧЕНИКИ
* Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Nested Loop Join по атрибуту ЧЛВК\_ИД

**План 2**

* Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию Н\_ЛЮДИ.ФАМИЛИЯ = 'Петров'
* Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409
* Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Hash Join по атрибуту ЧЛВК\_ИД
* Полный скан таблицы Н\_УЧЕНИКИ
* Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИКИ с использованием Hash Join по атрибуту ЧЛВК\_ИД

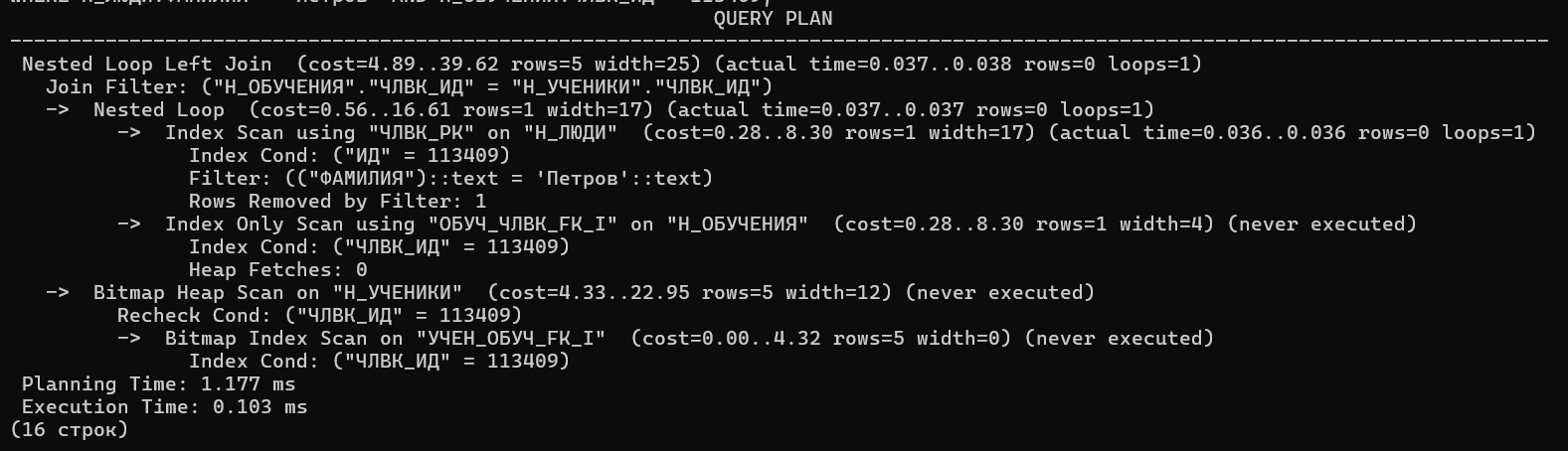
### ОПТИМАЛЬНЫЙ ПЛАН:

План 1, так как Nested Loop Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие и заранее известно, что результат фильтрации будет содержать малое количество строк



### ПРИ ДОБАВЛЕНИИ ИНДЕКСОВ ПЛАН ВЫПОЛНЕНИЯ ЗАПРОСА ИЗМЕНИТСЯ:

* Вместо полного скана таблиц будет использоваться индексный скан
* Nested Loop Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД



EXPLAIN ANALYZE SELECT Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО

FROM Н\_ЛЮДИ

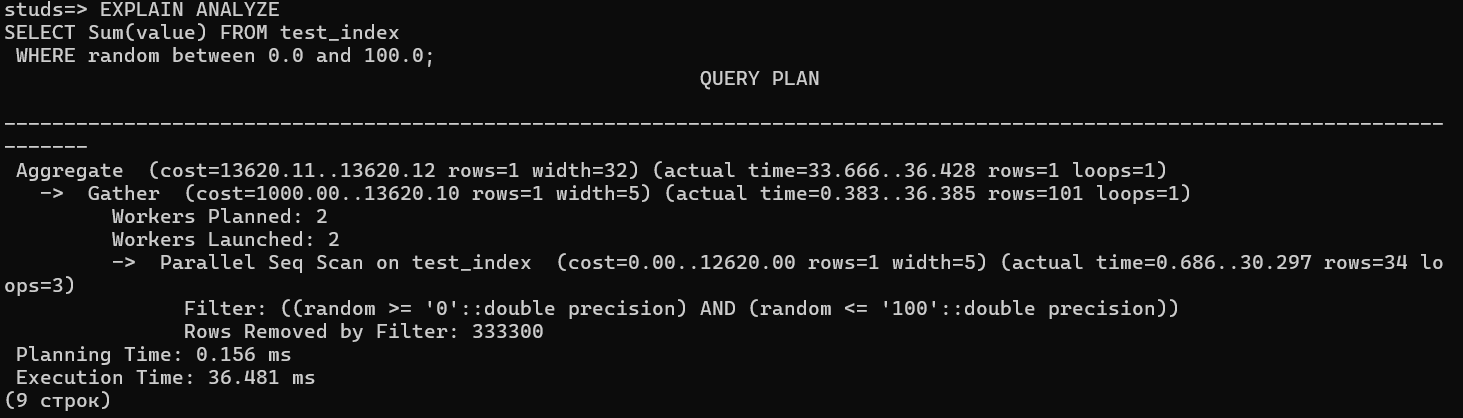
LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

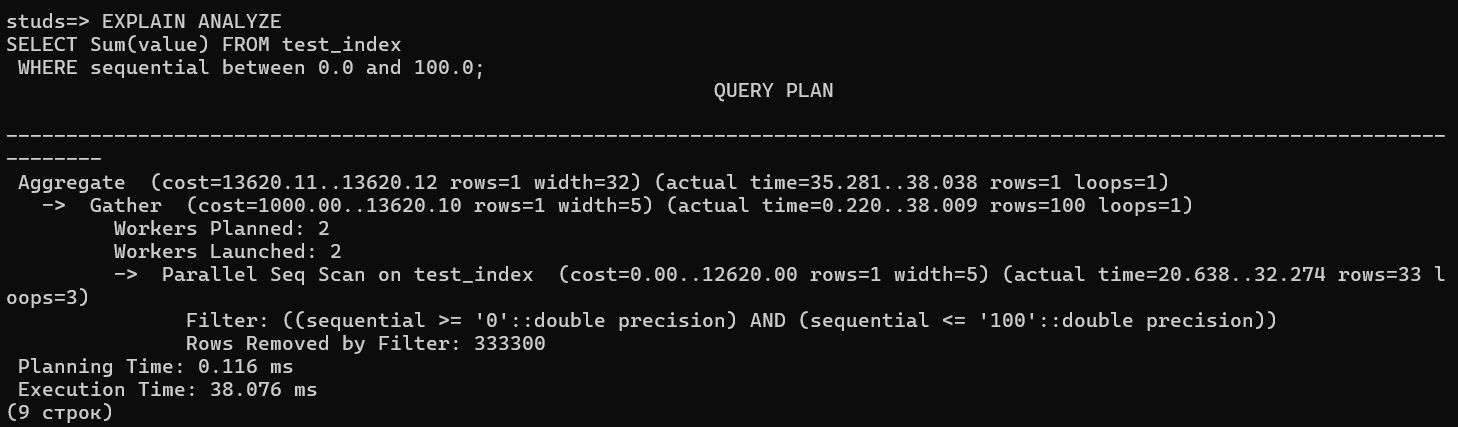
LEFT JOIN Н\_УЧЕНИКИ ON Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

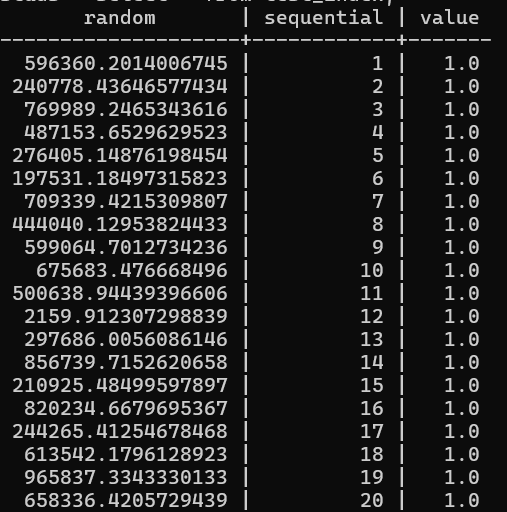
WHERE Н\_ЛЮДИ.ФАМИЛИЯ = 'Петров' AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 113409;

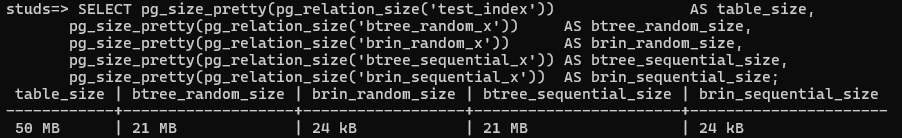
## Вывод:

В ходе выполнения лабораторной работы я освоила работу с реляционной алгеброй и научилась строить план выполнения запросов, а также их диаграммы. Изучила различные виды индексов и узнала, как использовать их для оптимизации запросов.

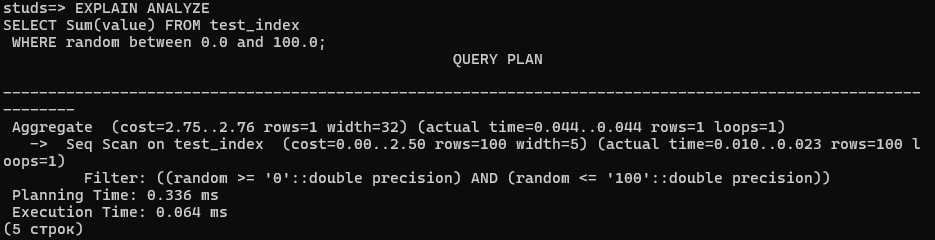


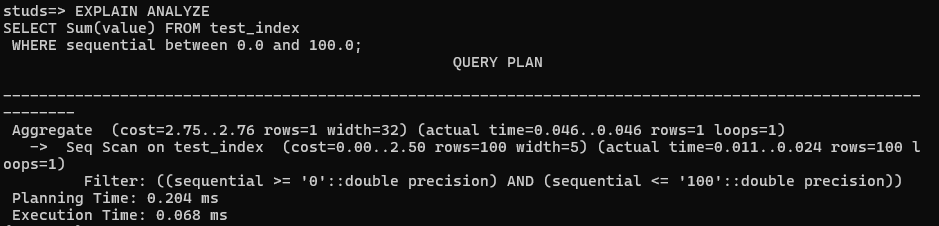




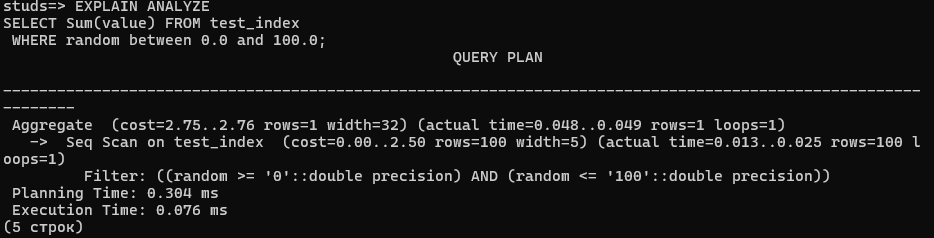


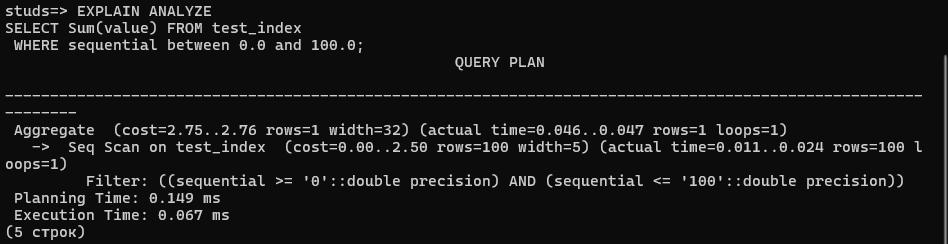
100 строк, btree

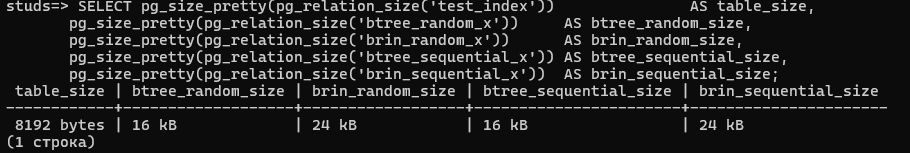




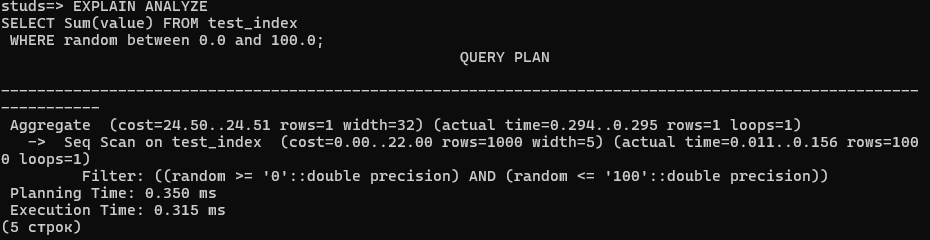
Brin

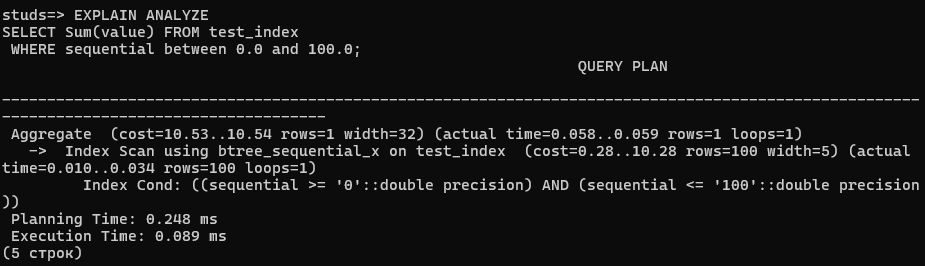




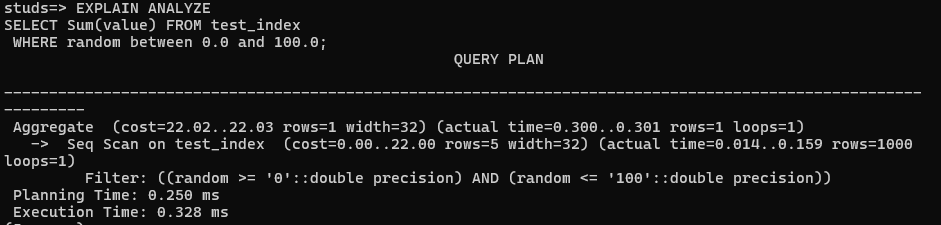


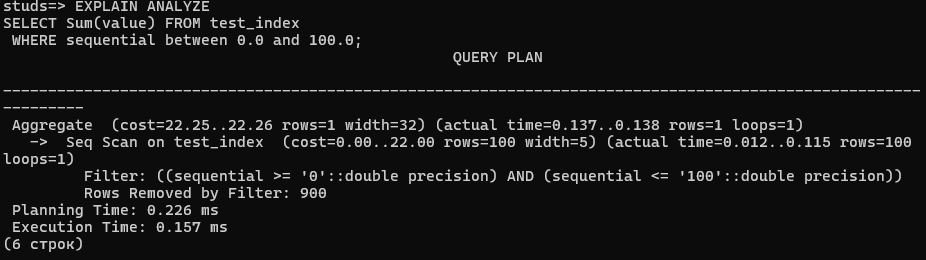
1000 строк, btree

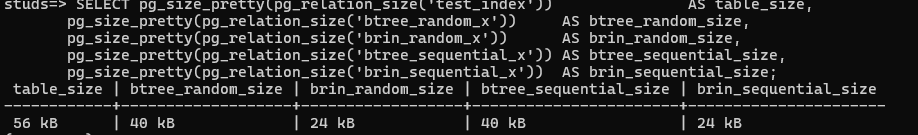




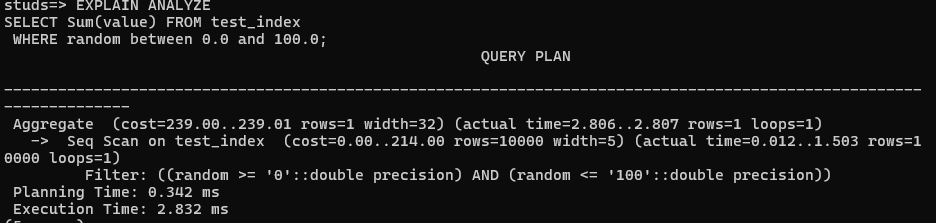
Brin

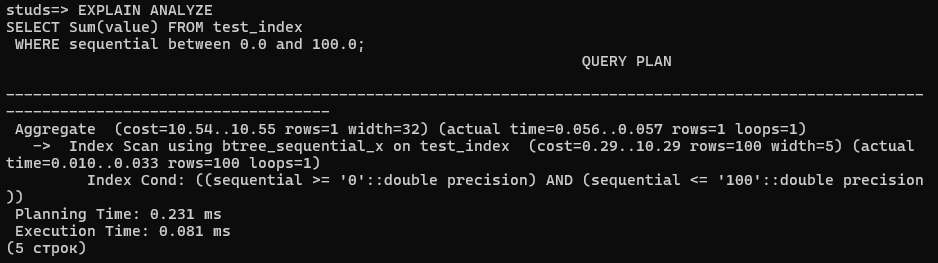




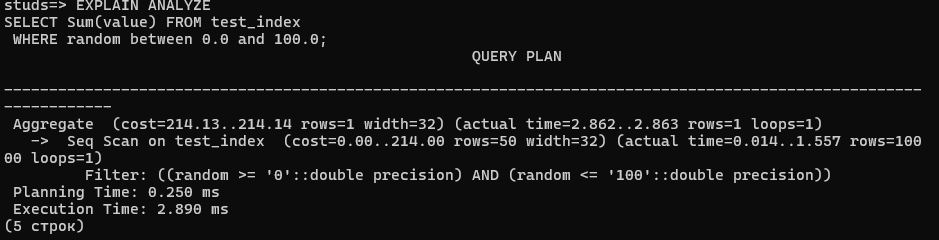


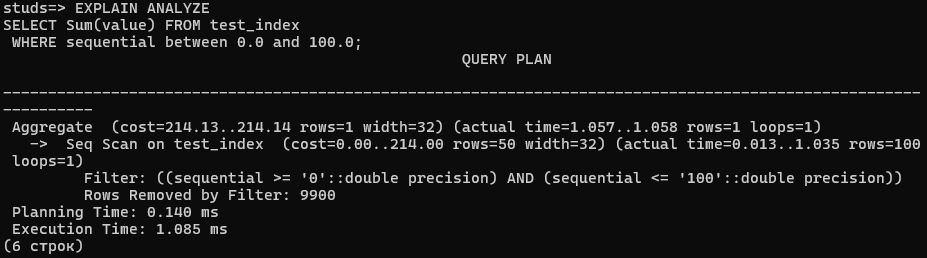
10000 строк, btree

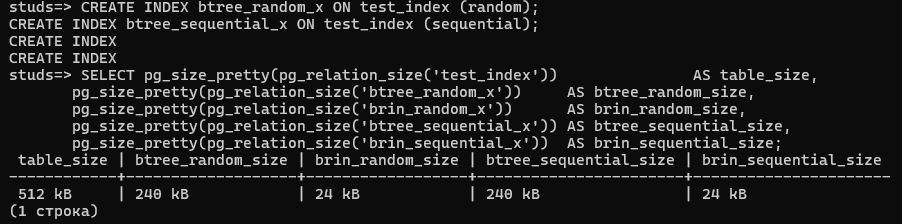




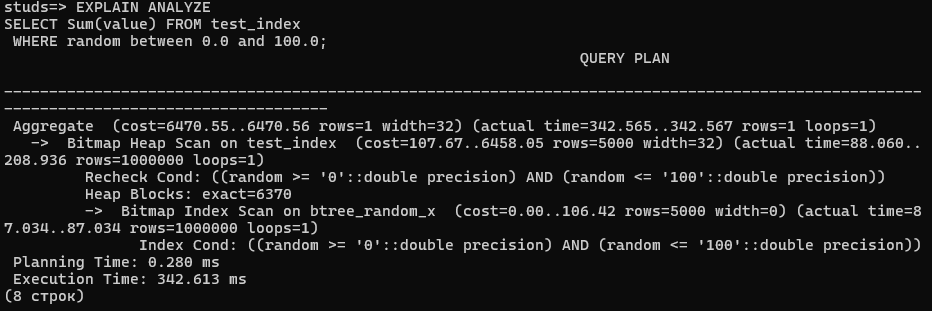
Brin

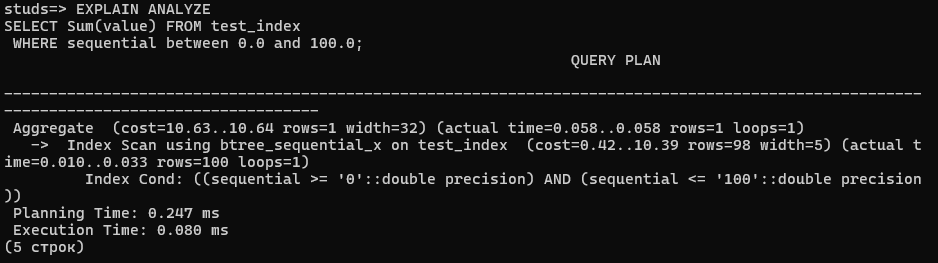




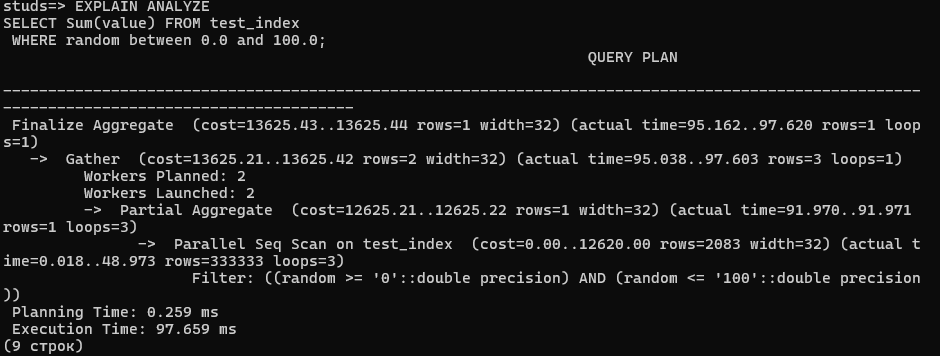


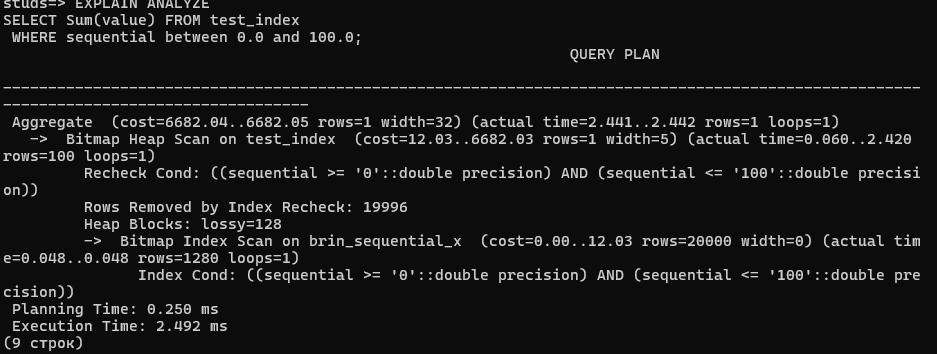
1000000, btree

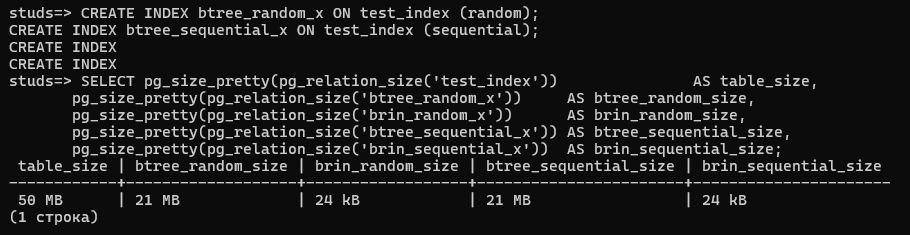


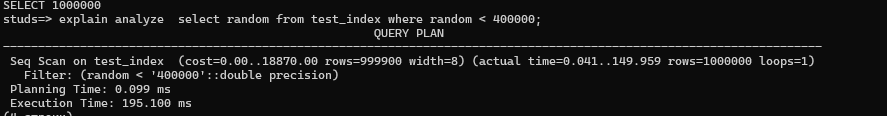


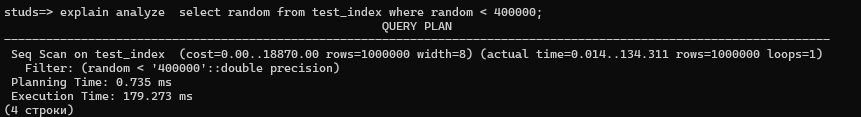
Brin

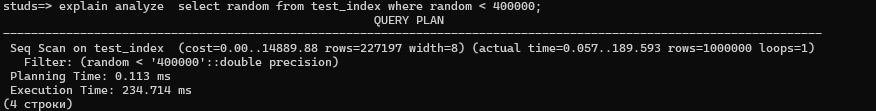




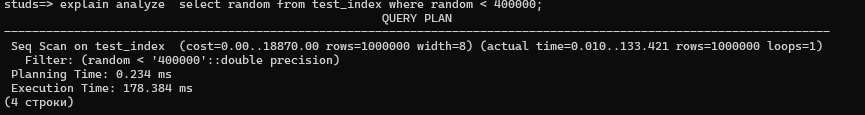








Btree



Btrin

