

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет программной инженерии и компьютерных технологий

Базы данных
Лабораторная работа № 4

Выполнил студент

Шнейдерис Герардас

Группа № Р3120

Преподаватель:

Николаев Владимир Вячеславович

г. Санкт-Петербург

2024 г.

Лабораторная работа № 3	1
Задание:	3
Ход работы:	4
Зависимости в схеме:	4
Отчет:	5
Вывод:	6
Список литературы:	6

Вариант:

6532

Задание:

1. Составить запросы на языке SQL (пункты 1-2).
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).
3. 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.
4. Изменяются ли планы при добавлении индекса и как?
5. Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]
6. Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете (планы выполнения запросов должны быть нарисованы, ответы на вопросы - представлены в текстовом виде).

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

- Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
- Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД.
- Фильтры (AND):
 - а) Н_ЛЮДИ.ИД < 100865.
 - б) Н_ВЕДОМОСТИ.ДАТА = 2022-06-08.
 - в) Н_ВЕДОМОСТИ.ДАТА > 2022-06-08.
- Вид соединения: LEFT JOIN.

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

- Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.
- Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ДАТА.
- Фильтры (AND):
 - а) Н_ЛЮДИ.ИД < 152862.
 - б) Н_ВЕДОМОСТИ.ДАТА = 2022-06-08.
- Вид соединения: LEFT JOIN.

Ход работы:

Запросы без индексов:

```
C: > Users > brain > .vscode > lab4.sql
1  -- 1 without indexes
2  SELECT Н_ЛЮДИ.ИД AS "id человека", Н_ВЕДОМОСТИ.ЧЛВК_ИД AS "id в ведомости"
3  FROM Н_ЛЮДИ
4  LEFT JOIN Н_ВЕДОМОСТИ ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
5  WHERE
6      Н_ЛЮДИ.ИД > 100865 AND
7      Н_ВЕДОМОСТИ.ДАТА < '2022-06-08';
8
9  -- 2 without indexes
10 SELECT
11     Н_ЛЮДИ.ИД AS "id человека",
12     Н_ВЕДОМОСТИ.ИД AS "id ведомости",
13     Н_СЕССИЯ.ДАТА AS "дата сессии"
14 FROM Н_ЛЮДИ
15 LEFT JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
16 LEFT JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
17 WHERE
18     Н_ЛЮДИ.ИД < 152862 AND
19     Н_ВЕДОМОСТИ.ДАТА = '2022-06-08';
```

Возможные индексы:

1

Можно сделать индекс для атрибута Н_ВЕДОМОСТИ.ЧЛВК_ИД (тип btree)

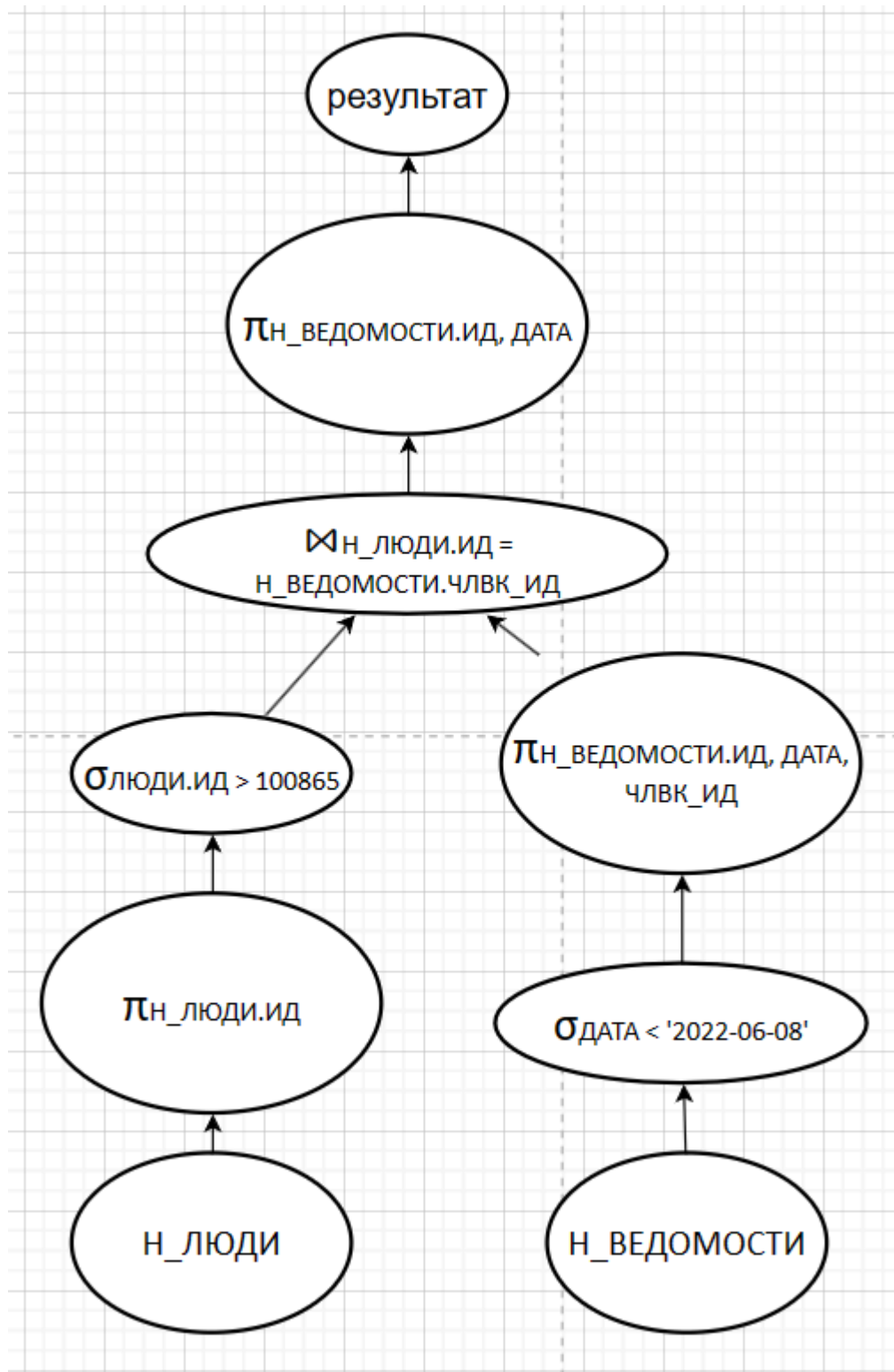
В тета-соединении поиск ведомости по ее id будет быстрее с использованием В-дерева

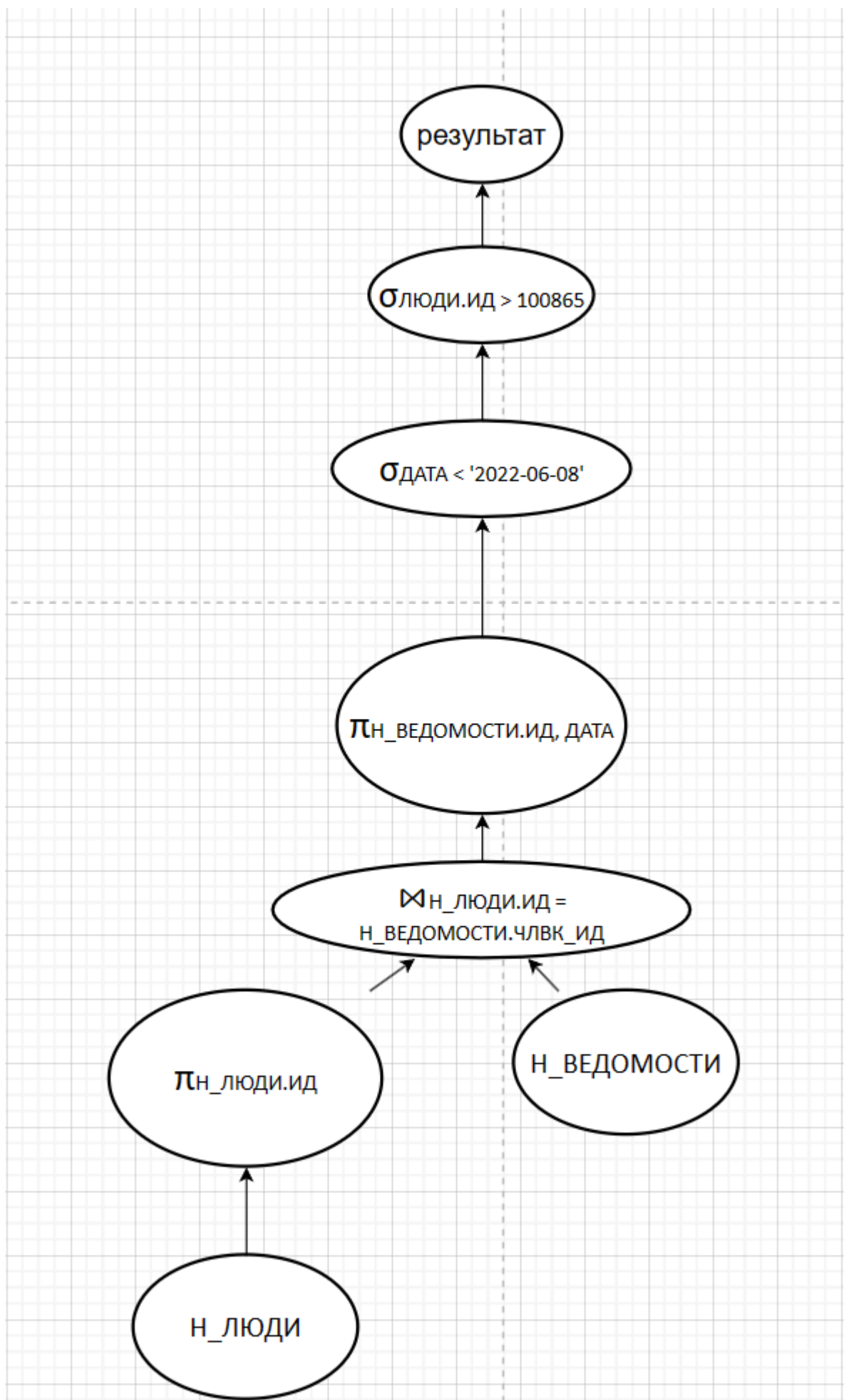
Допустимо создание индексов для атрибутов Н_ЛЮДИ.ИД и Н_ВЕДОМОСТИ.ДАТА(тип btree), которые бы ускорили проверку условий к соответствующим столбцам.

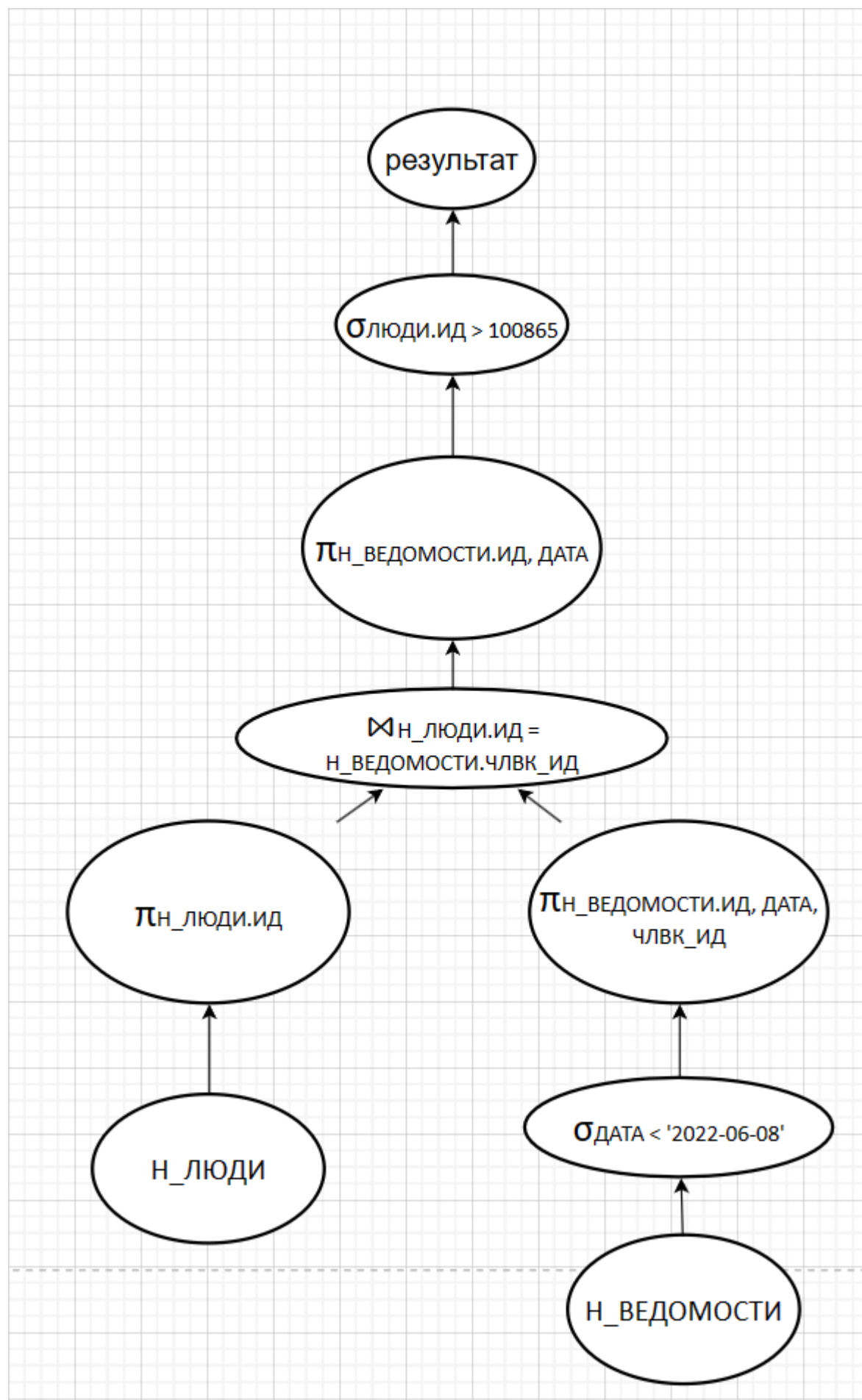
2

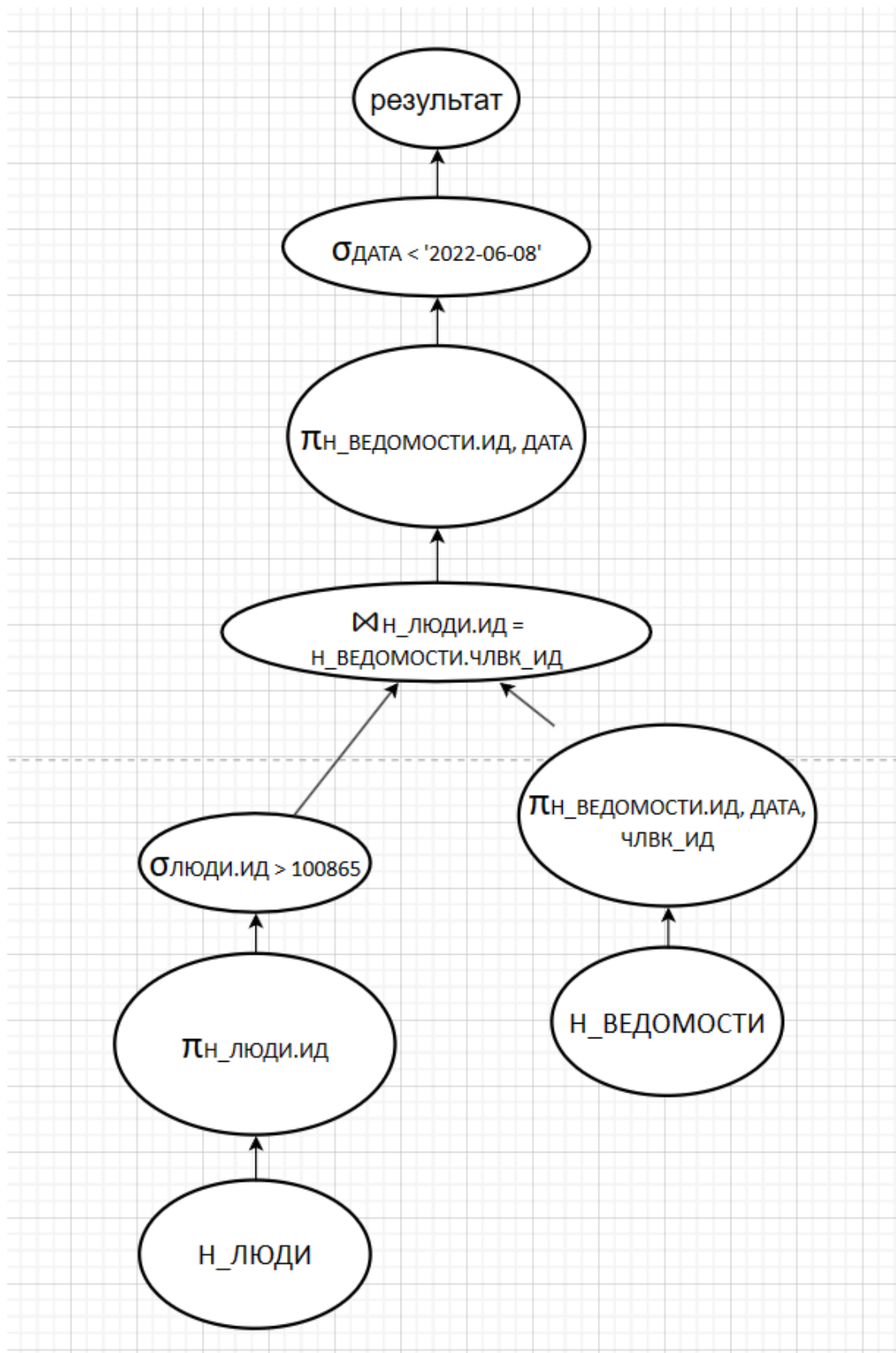
Аналогично, можно сделать btree индекс для Н_ЛЮДИ.ИД, чтобы быстро определять диапазон нужных id. Также можно сделать btree индексы для Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.ЧЛВК_ИД они бы ускорили соответствующие операции соединения. На Н_ВЕДОМОСТИ.ДАТА лучше сделать hash-index т.к. он будет эффективен для проверки условия на равенство.

Планы к запросу 1:



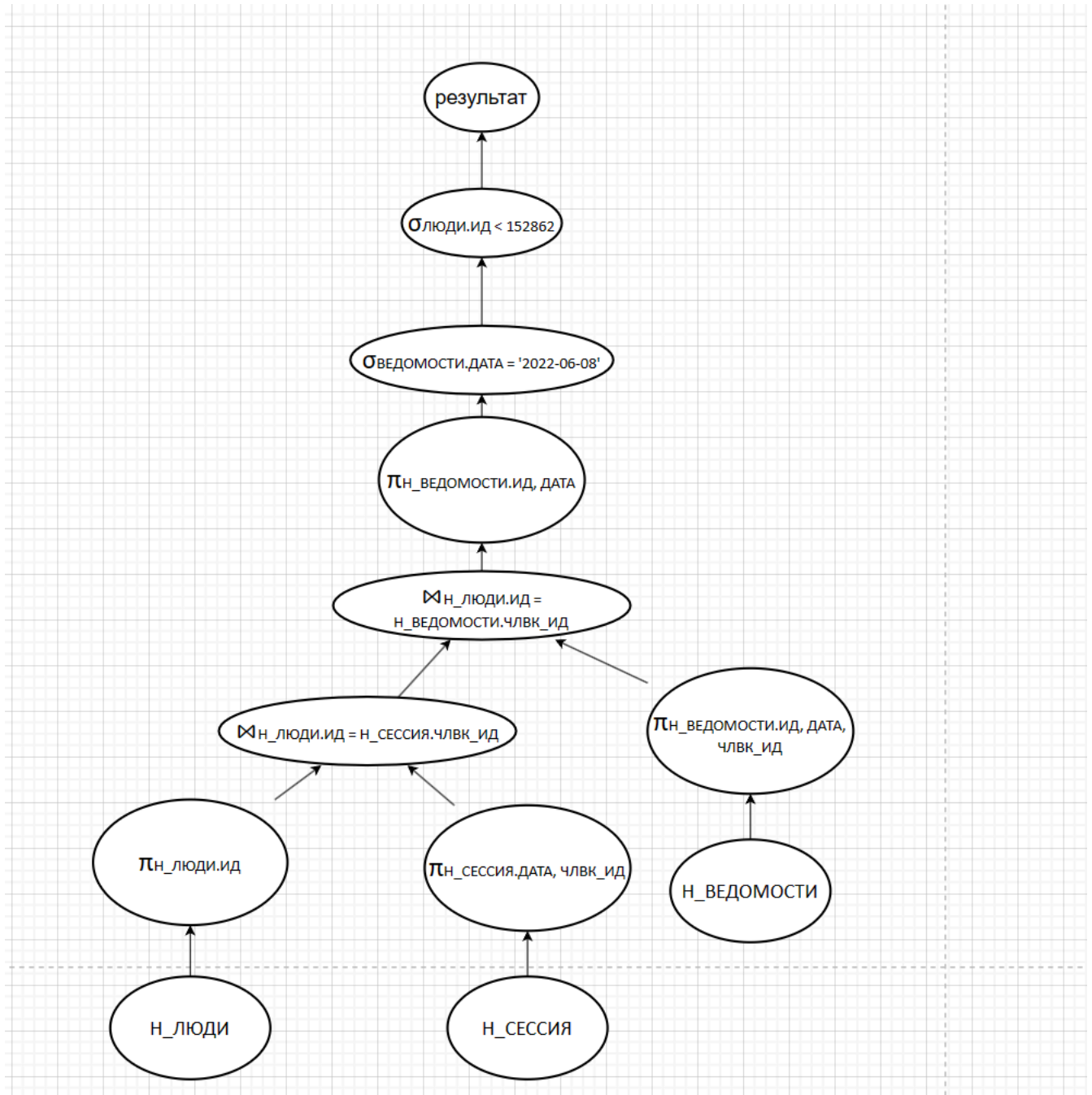


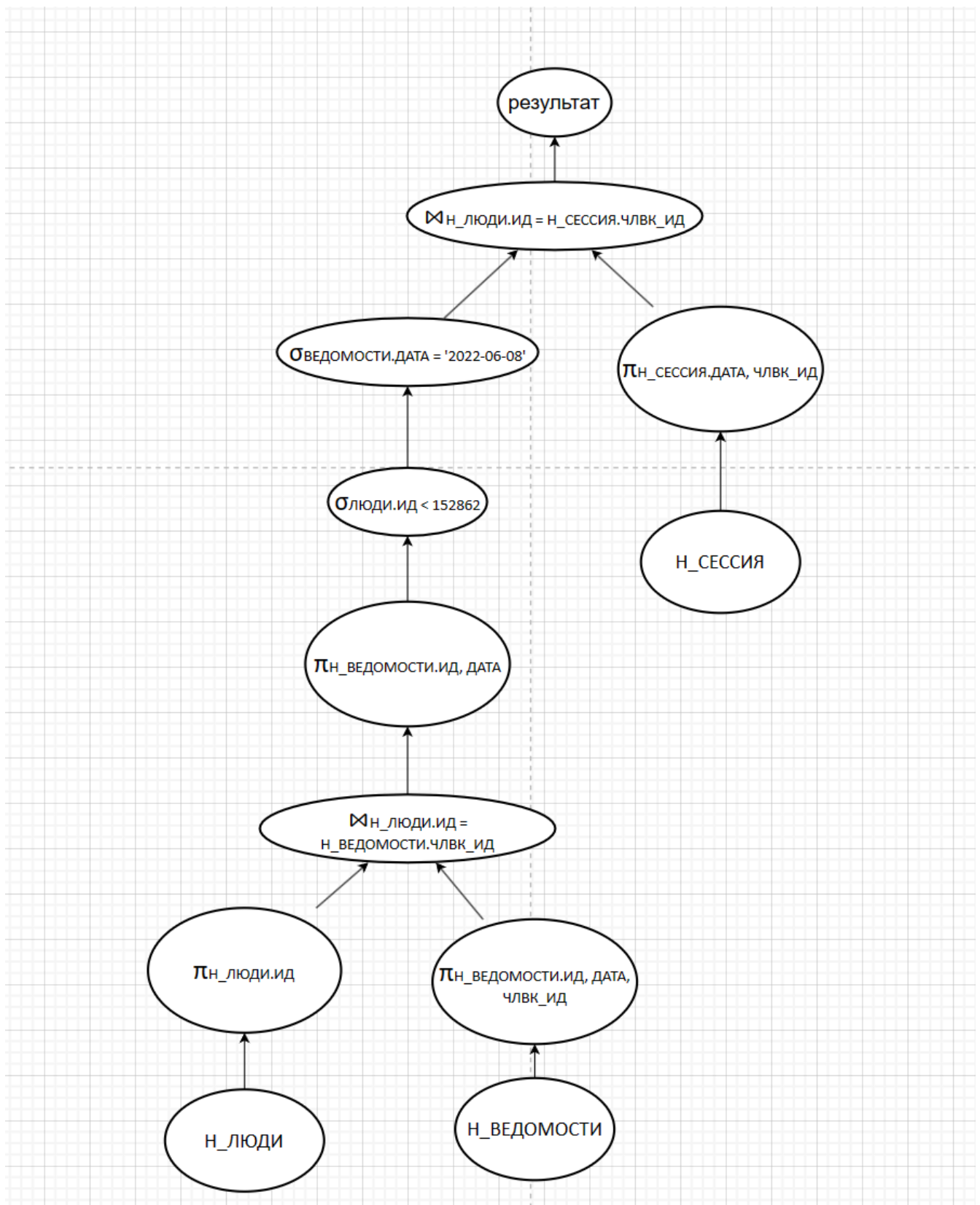




Оптимальный на мой взгляд последний план т.к. это левостороннее дерево позволяющее конвейерную обработку данных.

Планы к запросу 2:







Оптимальный на мой взгляд последний план т.к. это левостороннее дерево позволяющее конвейерную обработку данных.

Влияние индексов на планы:

Предложенные мной индексы не оказали бы значительного влияния на планы: проверка условий проводилась бы не по атрибутам, а по их индексам (что уменьшило бы время выполнения запроса), а для соединения таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ (и Н_СЕССИЯ во 2ом запросе)использовался бы индекс Н_ВЕДОМОСТИ.ЧЛВК_ИД(Н_СЕССИЯ.ЧЛВК_ИД). По сути последовательность действий осталась такой же, но времени каждое занимало б меньше.

Вывод explain analyze:

QUERY PLAN

```
Nested Loop Left Join (cost=0.86..411.52 rows=60 width=16) (actual time=0.083..0.094 rows=3 loops=1)
-> Nested Loop (cost=0.58..360.72 rows=60 width=8) (actual time=0.070..0.077 rows=3 loops=1)
    -> Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" (cost=0.29..214.94 rows=72 width=8) (actual time=0.058..0.060 rows=3 loops=1)
        Index Cond: ("ДАТА" = '2022-06-08 00:00:00'::timestamp without time zone)
    -> Index Only Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..2.02 rows=1 width=4) (actual time=0.003..0.003 rows=1 loops=3)
        Index Cond: (("ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД") AND ("ИД" < 152862))
        Heap Fetches: 0
-> Index Scan using "SYS_C003500_IFK" on "Н_СЕССИЯ" (cost=0.28..0.67 rows=18 width=12) (actual time=0.004..0.004 rows=0 loops=3)
    Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
Planning Time: 1.706 ms
Execution Time: 0.205 ms
(11 строк)
```

QUERY PLAN

```
Hash Join (cost=277.01..7707.71 rows=221195 width=8) (actual time=1.570..98.780 rows=222437 loops=1)
Hash Cond: ("Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
-> Seq Scan on "Н_ВЕДОМОСТИ" (cost=0.00..6846.50 rows=222368 width=4) (actual time=0.014..46.377 rows=222437 loops=1)
    Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)
    Rows Removed by Filter: 3
-> Hash (cost=213.38..213.38 rows=5091 width=4) (actual time=1.498..1.500 rows=5077 loops=1)
    Buckets: 8192 Batches: 1 Memory Usage: 243kB
    -> Index Only Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..213.38 rows=5091 width=4) (actual time=0.008..0.704 rows=5077 loops=1)
        Index Cond: ("ИД" > 100865)
        Heap Fetches: 0
Planning Time: 0.373 ms
Execution Time: 109.211 ms
(12 строк)
```

Отчет:

Ссылка на репозиторий - [GeruniaSun/ITMO-DB-lab4 \(github.com\)](https://github.com/GeruniaSun/ITMO-DB-lab4)

Вывод:

В рамках выполнения этой работы я узнал о такой важной области работы с базами данных, как выполнение и оптимизация запросов. Я узнал о планах выполнения, индексах различных видов и нескольких алгоритмах JOIN. Помимо этого я узнал о команде EXPLAIN и научился читать её вывод.

Список литературы:

В.В. Кириллов, Г.Ю. Громов - Введение в реляционные базы данных

[PostgreSQL: Documentation: 16: PostgreSQL 16.2 Documentation](#)

[SQL Syntax Checker And SQL Syntax Validator | CodersTool](#)

[Учебник по языку SQL \(DDL, DML\)/ Хабр](#)