

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.01 Информатика и вычислительная
техника

Дисциплина «Базы данных»

Лабораторная работа №4

Вариант 3422

Студент

Баянов Р. Д.

P3134

Преподаватель

Перцев Т.

Санкт-Петербург

2023 г.

Оглавление

Задание	3
Вариант.....	3
Реализация запросов на SQL	4
Планы выполнения запросов	7
Вывод команды EXPLAIN ANALYZE.....	14
Вывод	15

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.

Изменяются 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).

Вариант

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ИД, Н_СЕССИЯ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ИМЯ < Александр.
б) Н_СЕССИЯ.ДАТА = 2002-01-04.
Вид соединения: LEFT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.УЧГОД.
Фильтры (AND):
а) Н_ЛЮДИ.ОТЧЕСТВО > Сергеевич.
б) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.
с) Н_СЕССИЯ.ИД < 1975.
Вид соединения: INNER JOIN.

Реализация запросов на SQL

```
SELECT "Н_люди"."ИД", "Н_сессия"."ДАТА"
FROM "Н_люди"
      LEFT JOIN "Н_сессия" ON "ЧЛВК_ИД" = "Н_люди"."ИД"
WHERE "Н_люди"."ИМЯ" < 'Александр'
      AND "Н_сессия"."ДАТА" = '2002-01-04';

SELECT "Н_люди"."ИМЯ", "Н_ведомости"."ИД", "Н_сессия"."УЧГОД"
FROM "Н_люди"
      INNER JOIN "Н_сессия" ON "ЧЛВК_ИД" = "Н_люди"."ИД"
      INNER JOIN "Н_ведомости" ON "Н_люди"."ИД" = "Н_ведомости"."ЧЛВК_ИД"
WHERE "Н_люди"."ОТЧЕСТВО" > 'Сергеевич'
      AND "Н_сессия"."ИД" = '1975'
      AND "Н_ведомости"."ДАТА" < '2022-06-08';
```

Индексирование

Первый запрос:

1) CREATE INDEX ЛЮДИ_ИМЯ ON “Н_ЛЮДИ” USING btree(“ИМЯ”);

1. Запрос выдаёт несколько строк, а не одну строку.
2. Данный атрибут используется в фильтрации с помощью WHERE.
3. Запрос использует оператор “<”.
4. Данный атрибут, вероятно, почти не будет изменяться, а значит, индекс не нужно будет обновлять часто.

2) CREATE INDEX СЕССИЯ_ДАТА ON “Н_СЕССИЯ” USING hash(“ДАТА”);

1. Запрос выдаёт несколько строк, а не одну строку.
2. Данный атрибут используется в фильтрации с помощью WHERE.
3. Запрос использует оператор “=”.
4. Данный атрибут, вероятно, почти не будет изменяться, а значит, индекс не нужно будет обновлять часто.
5. Нет выборки большого объёма данных, так как сессия получается только за один день.

3) CREATE INDEX СЕССИЯ_ЧЛБК_ИД ON “Н_СЕССИЯ” USING hash(“ЧЛБК_ИД”)

1. В операции участвует соединение таблиц.
2. Столбец используется в JOIN.
3. Используется прямое сравнение.

4) CREATE INDEX ЛЮДИ_ИД ON “Н_ЛЮДИ” USING hash(“ИД”)

1. В операции участвует соединение таблиц.
2. Данный столбец используется в JOIN.
3. Значения атрибута уникальны, так как являются первичным ключом.

Второй запрос:

1) CREATE INDEX ВЕДОМОСТИ_ДАТА ON “Н_ВЕДОМОСТИ” USING btree(ДАТА)

1. Запрос выдаёт несколько строк, а не одну строку.
2. Данный атрибут используется в фильтрации с помощью WHERE.
3. Запрос использует оператор “>”.

2) CREATE INDEX ЛЮДИ_ОТЧЕСТВО ON “Н_ЛЮДИ” USING
btree(ОТЧЕСТВО)

1. Запрос выдаёт несколько строк, а не одну строку.
2. Данный атрибут используется в фильтрации с помощью WHERE.
3. Данный атрибут, вероятно, почти не будет изменяться, а значит, индекс не нужно будет обновлять часто.

3) CREATE INDEX СЕССИЯ_ИД ON “Н_СЕССИЯ” USING hash(“ИД”)

1.

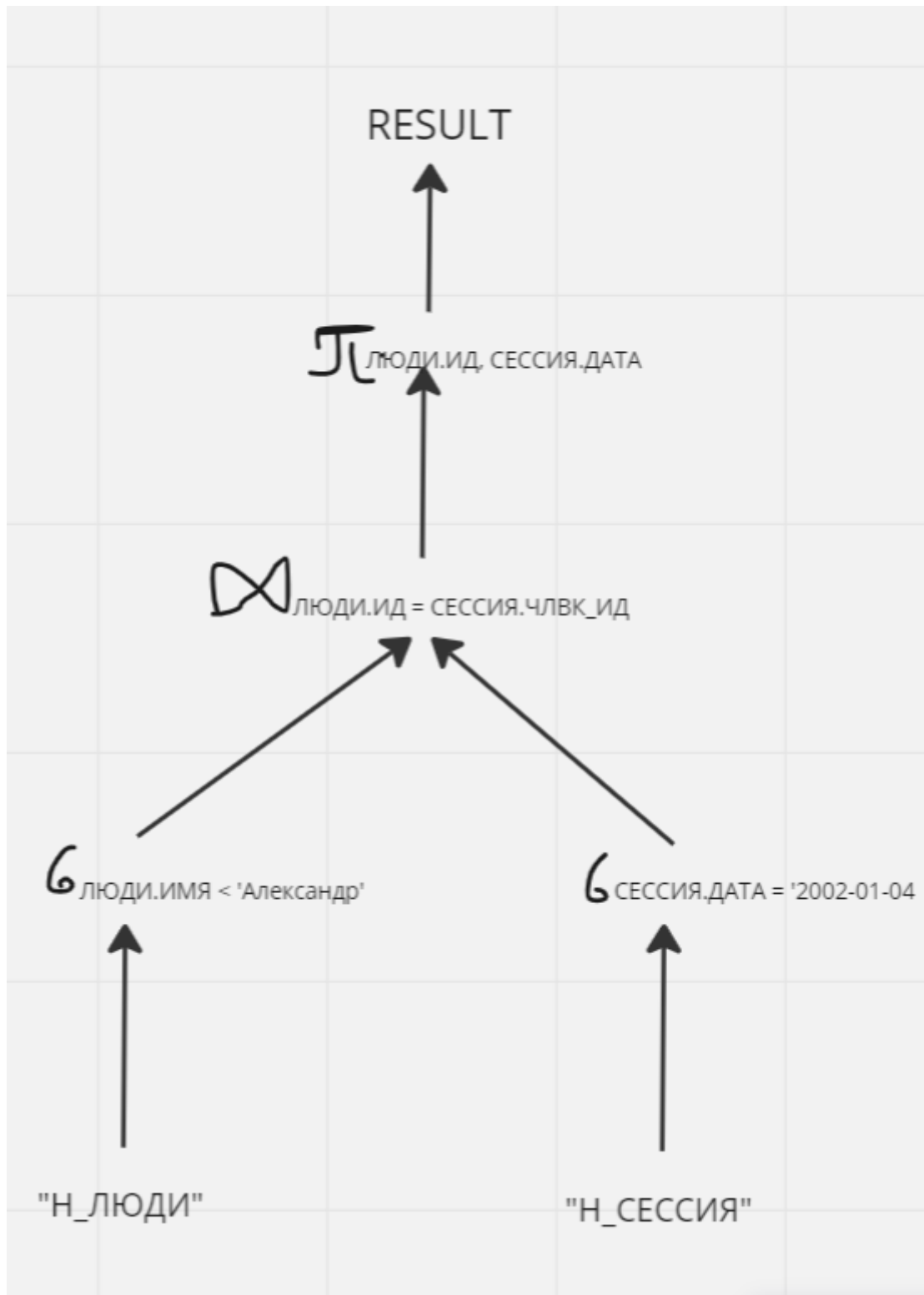
4) CREATE INDEX ЛЮДИ_ИД ON “Н_ЛЮДИ” USING hash(“ИД”)

5) CREATE INDEX СЕССИЯ_ЧЛБК_ИД ON “Н_СЕССИЯ” USING
hash(“ЧЛБК_ИД”)

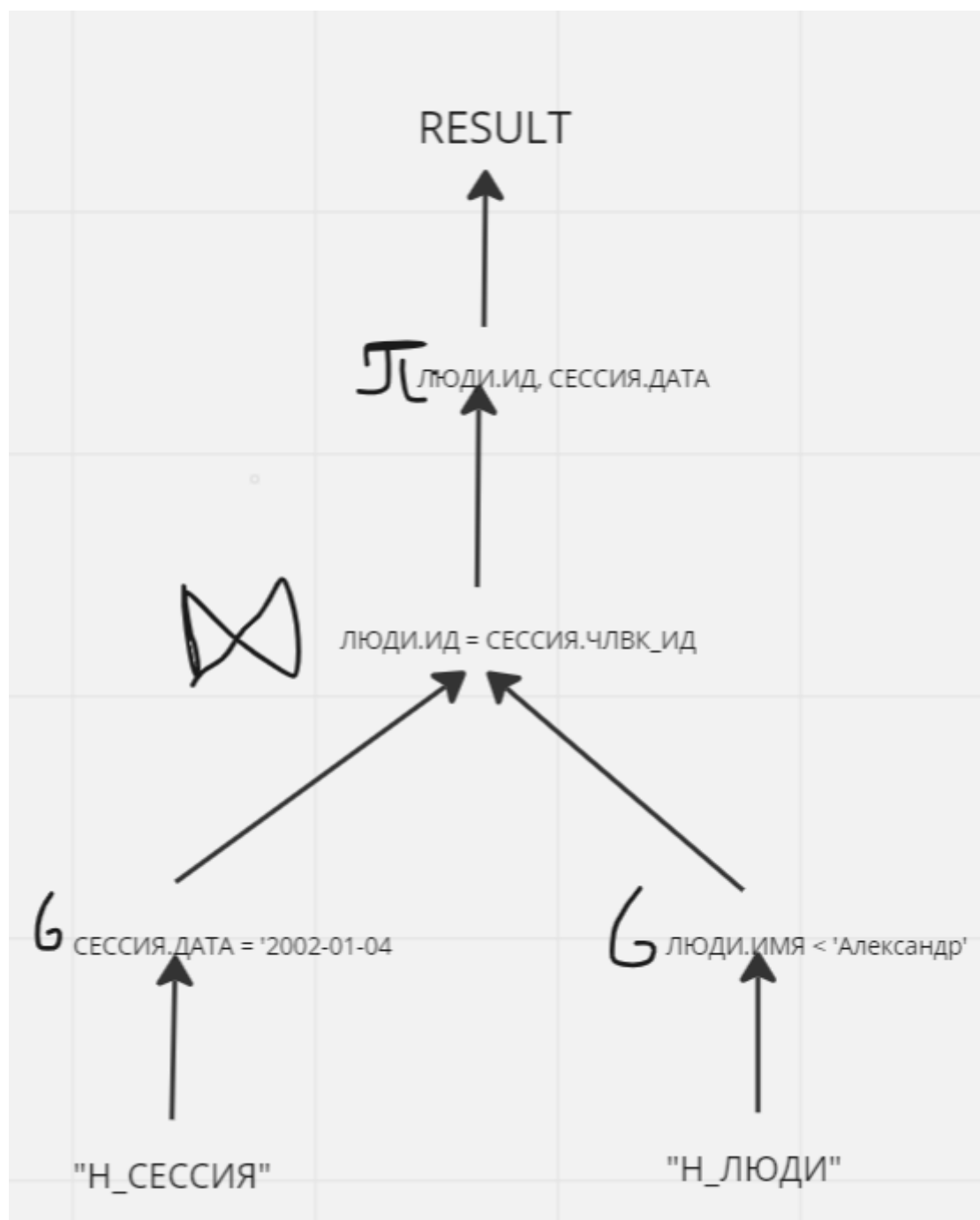
Планы выполнения запросов

Планы к 1 запросу:

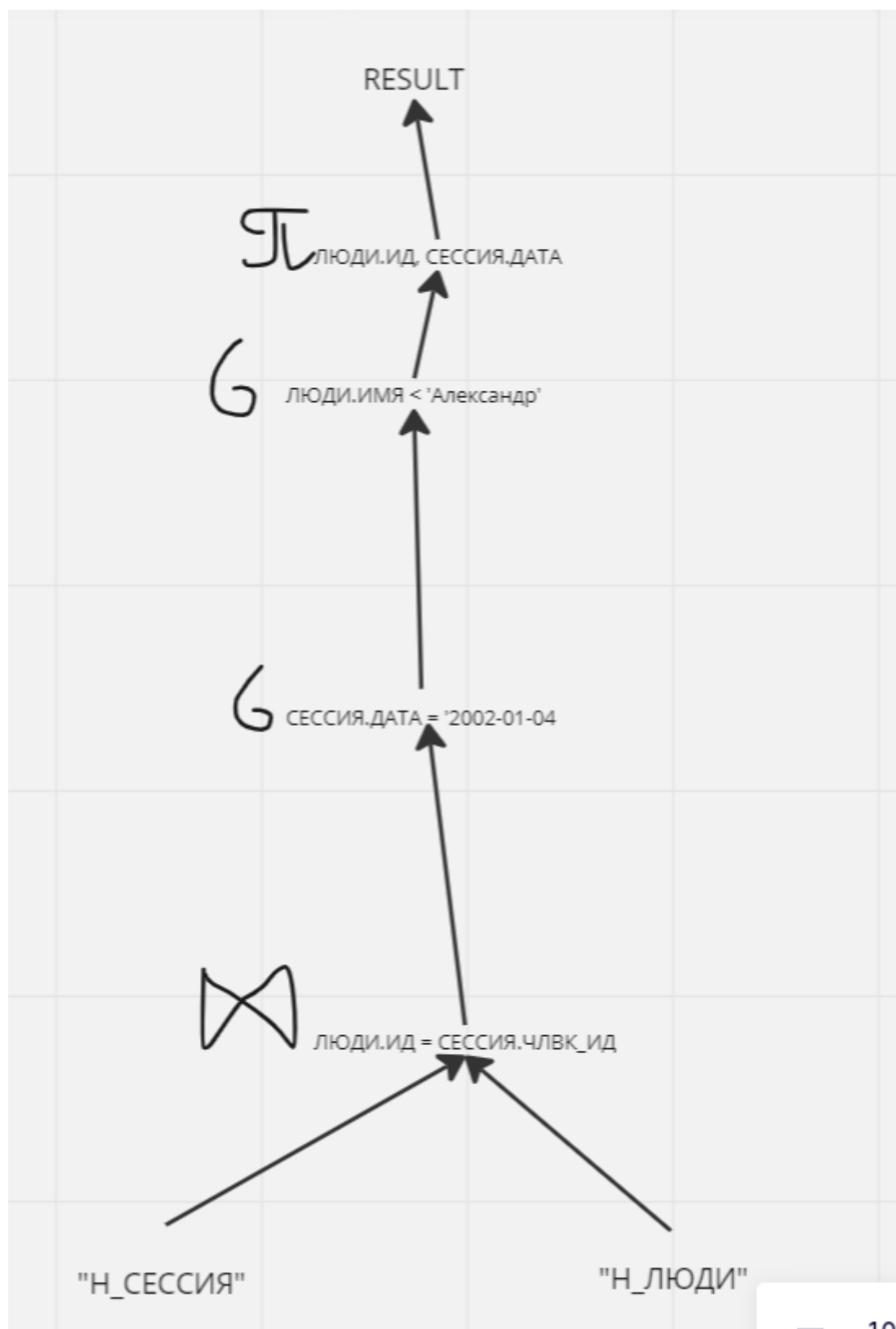
1)



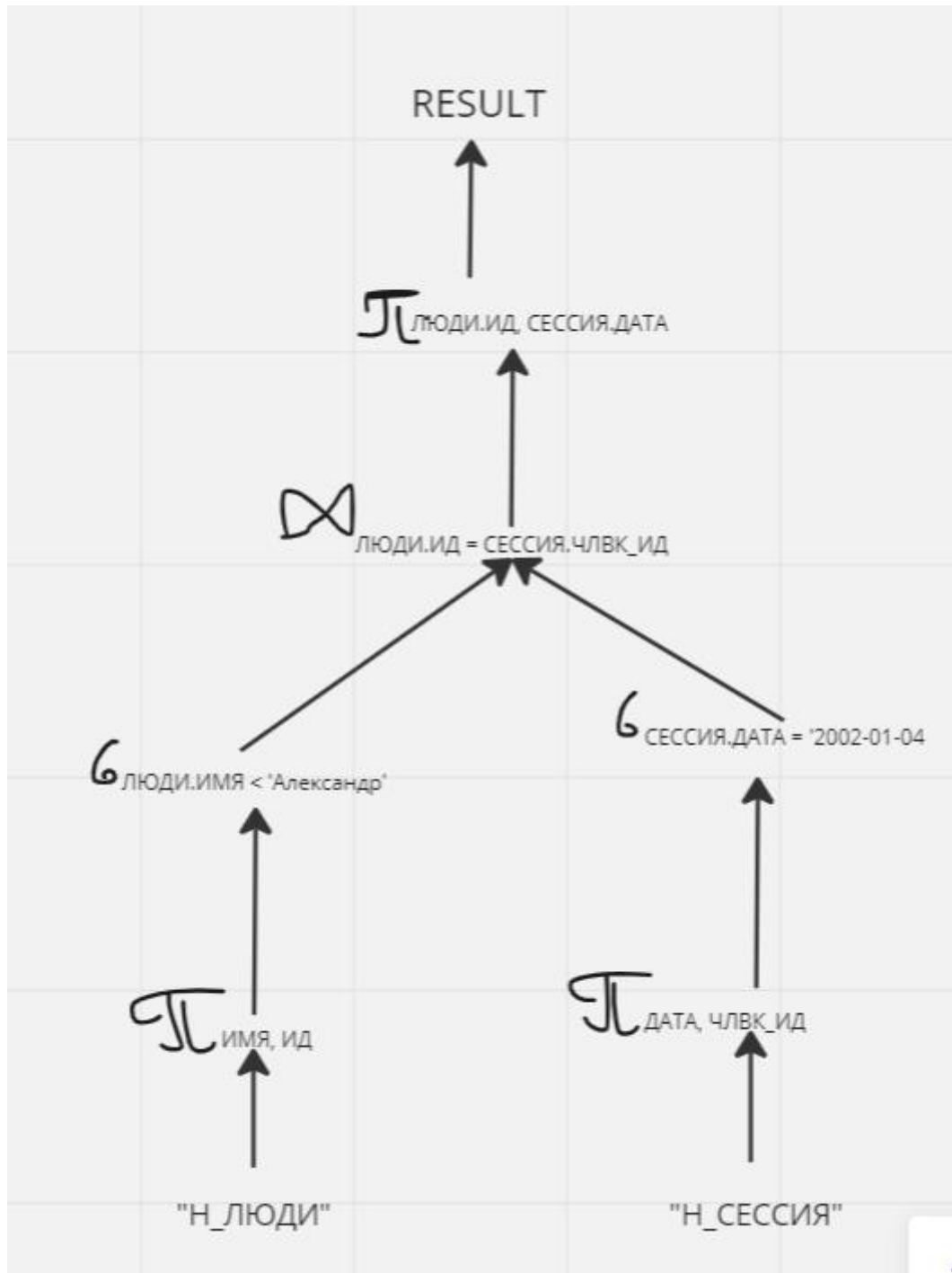
2)



3)



4)

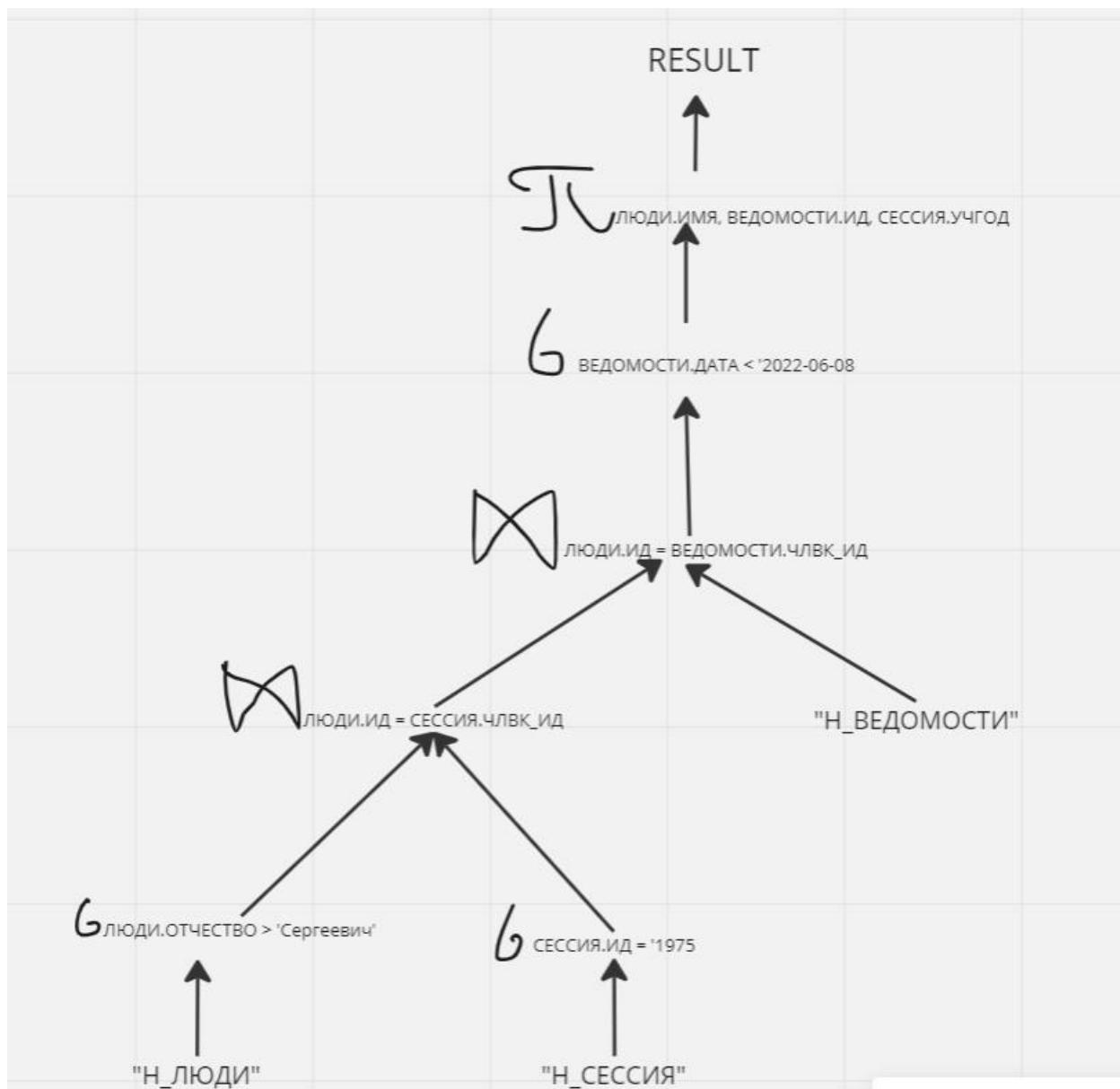


Последний план является оптимальным, так как:

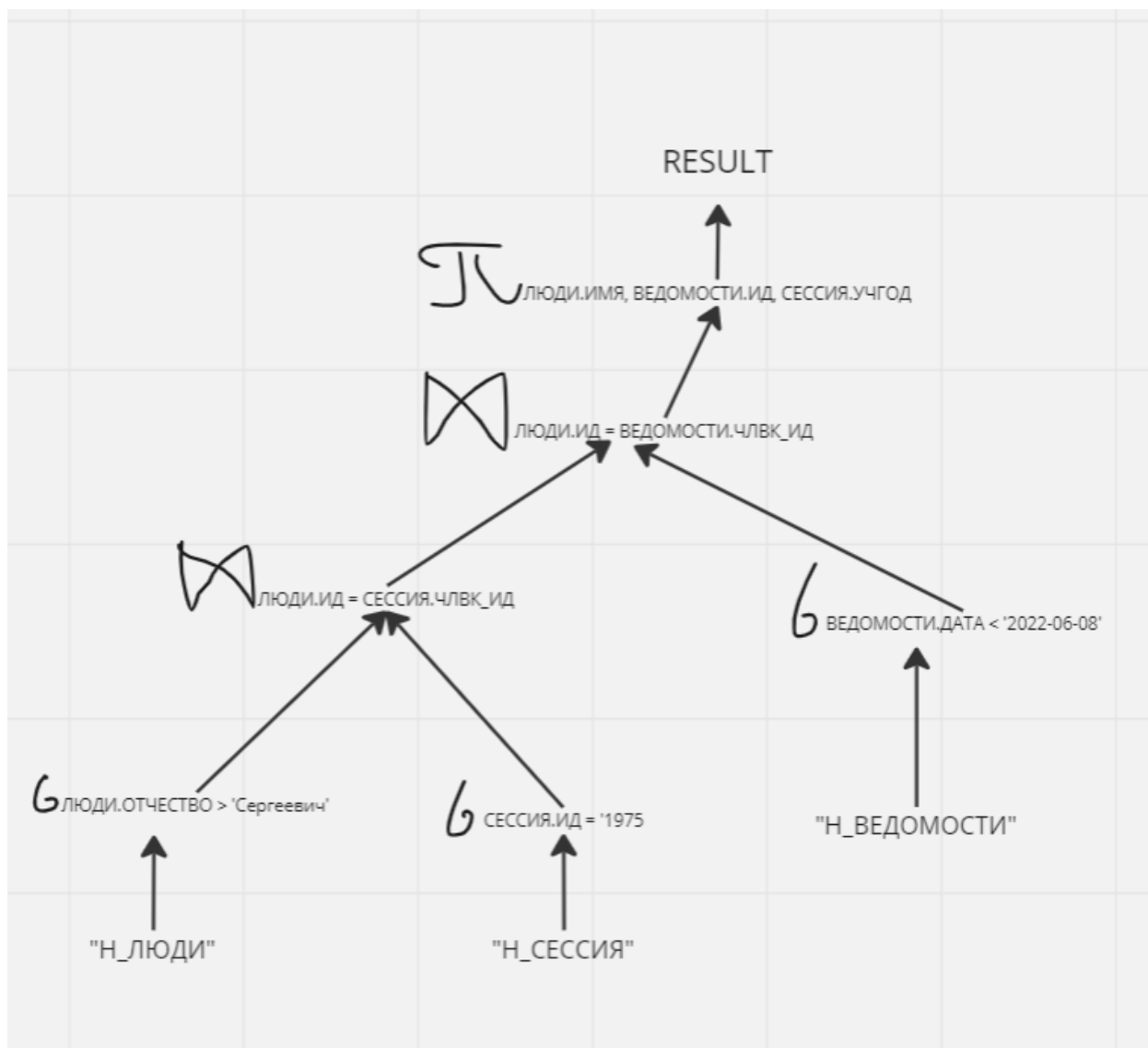
- 1) Проекция сделана раньше соединения;
- 2) Выборка сделана раньше соединения;
- 3) Минимизирован размер промежуточных данных.

Планы ко 2 запросу:

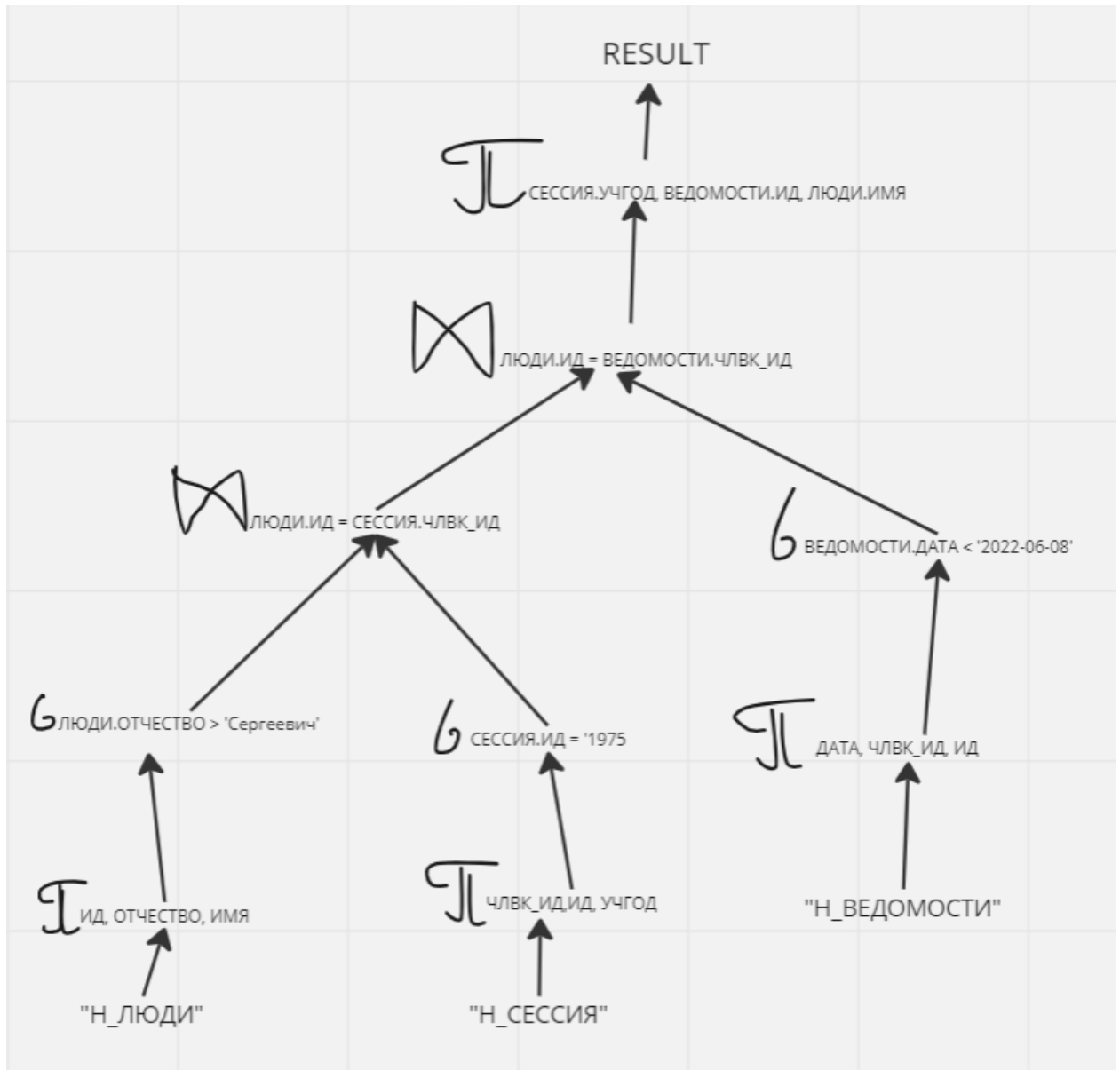
1)



2)



3)



Оптимальным является последний план, так как:

- 1) Проекция сделана раньше соединения;
- 2) Выборка сделана раньше соединения;
- 3) План левосторонний;
- 4) Минимизирован размер промежуточных данных.

Вывод команды EXPLAIN ANALYZE

Первый запрос:

QUERY PLAN	
1	Nested Loop (cost=0.28..159.50 rows=1 width=12) (actual time=0.511..0.512 rows=0 loops=1)
2	-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=5 width=12) (actual time=0.061..0.442 rows=7 loops=1)
3	Filter: ("Дата" = '2002-01-04 00:00:00'::timestamp without time zone)
4	Rows Removed by Filter: 3745
5	-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.009..0.009 rows=0 loops=7)
6	Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
7	Filter: (("Имя")::text < 'Александр'::text)
8	Rows Removed by Filter: 1
9	Planning Time: 1.140 ms
10	Execution Time: 0.559 ms

Второй запрос:

QUERY PLAN	
1	Nested Loop (cost=0.58..160.38 rows=56 width=12) (actual time=0.478..0.479 rows=0 loops=1)
2	Join Filter: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
3	-> Nested Loop (cost=0.28..126.24 rows=1 width=12) (actual time=0.478..0.479 rows=0 loops=1)
4	-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=5 width=12) (actual time=0.061..0.442 rows=7 loops=1)
5	Filter: ("ИД" = 1975)
6	Rows Removed by Filter: 3751
7	-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.009..0.009 rows=0 loops=7)
8	Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
9	Filter: (("ОТЧЕСТВО")::text > 'Александр'::text)
10	-> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.009..0.009 rows=0 loops=7)
11	Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
12	Filter: ("Дата" < '2022-06-08 00:00:00'::timestamp without time zone)
13	Planning Time: 0.706 ms
14	Execution Time: 0.478 ms

Вывод

При выполнении данной лабораторной работы я узнал понятие индексов в базах данных и как их использовать. Научился оптимизировать запросы и выбирать планы выполнения запросов.