

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

По «Базы Данных»

Вариант 1452

Выполнил:

Студент группы Р3106

Рубцов Арсений Дмитриевич

Проверил:

Гаврилов Антон Валерьевич

Оглавление

1.Текст задания.....	3
2.Реализация запросов на SQL.....	3
3.Уменьшение выполнения времени 1 запроса.....	4
Индексы, добавление которых уменьшит время выполнения.....	4
Возможные планы выполнения запросов без индексов.....	5
4. План выполнения запроса 1.....	6
5.Уменьшение выполнения времени 2 запроса.....	7
Индексы, добавление которых уменьшит время выполнения.....	7
Возможные планы выполнения запросов без индексов.....	7
6. План выполнения запроса 2.....	9
7.Вывод.....	11

1.Текст задания

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.

Изменяться 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).

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

Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 142390.

с) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.

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.ИМЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ЛЮДИ.ИД > 152862.

б) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.

Вид соединения: INNER JOIN.

2.Реализация запросов на SQL

--запрос 1

--Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по

указанным условиям:

--Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.

--Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД.

--Фильтры (AND):

--а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2.

--б) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 142390. ??

--с) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.

--Вид соединения: LEFT JOIN

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
LEFT JOIN Н_ВЕДОМОСТИ ON Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД =
Н_ВЕДОМОСТИ.ТВ_ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2 AND Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285;
```

--запрос 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.ЧЛВК_ИД.

--Фильтры (AND):

--а) Н_ЛЮДИ.ИД > 152862.

--б) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.

--Вид соединения: INNER JOIN.

```
SELECT Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ЧЛВК_ИД
FROM Н_ЛЮДИ
INNER JOIN Н_ВЕДОМОСТИ ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
INNER JOIN Н_СЕССИЯ ON Н_СЕССИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
WHERE Н_ЛЮДИ.ИД > 152862 AND Н_ВЕДОМОСТИ.ДАТА < '2022-06-08';
```

3. Уменьшение выполнения времени 1 запроса

Индексы, добавление которых уменьшит время выполнения

1) Н_ТИПЫ_ВЕДОМОСТЕЙ

- Индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2.

2) Н_ВЕДОМОСТИ

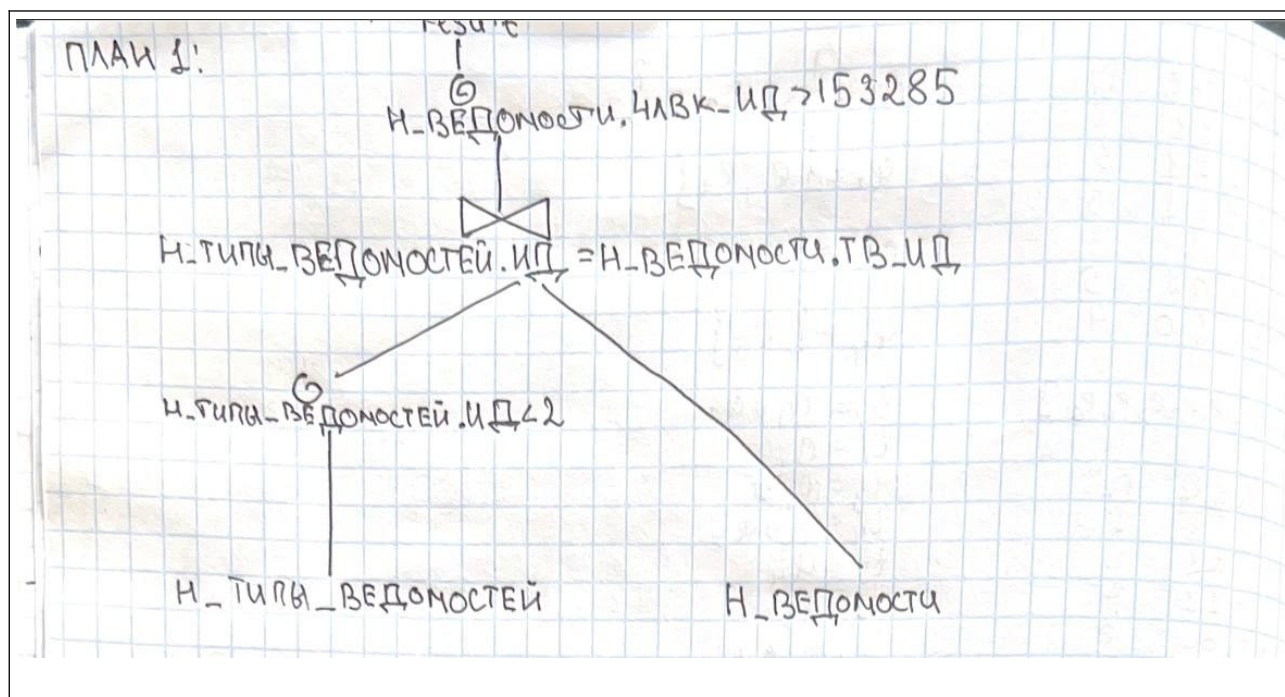
- Индекс на атрибуте ЧЛВК_ИД (B-tree). Это ускорит фильтрацию строк с условием Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285.
- Индекс на атрибуте ТВ_ИД (B-tree). Это ускорит операции JOIN по атрибуту ТВ_ИД.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запросов без индексов

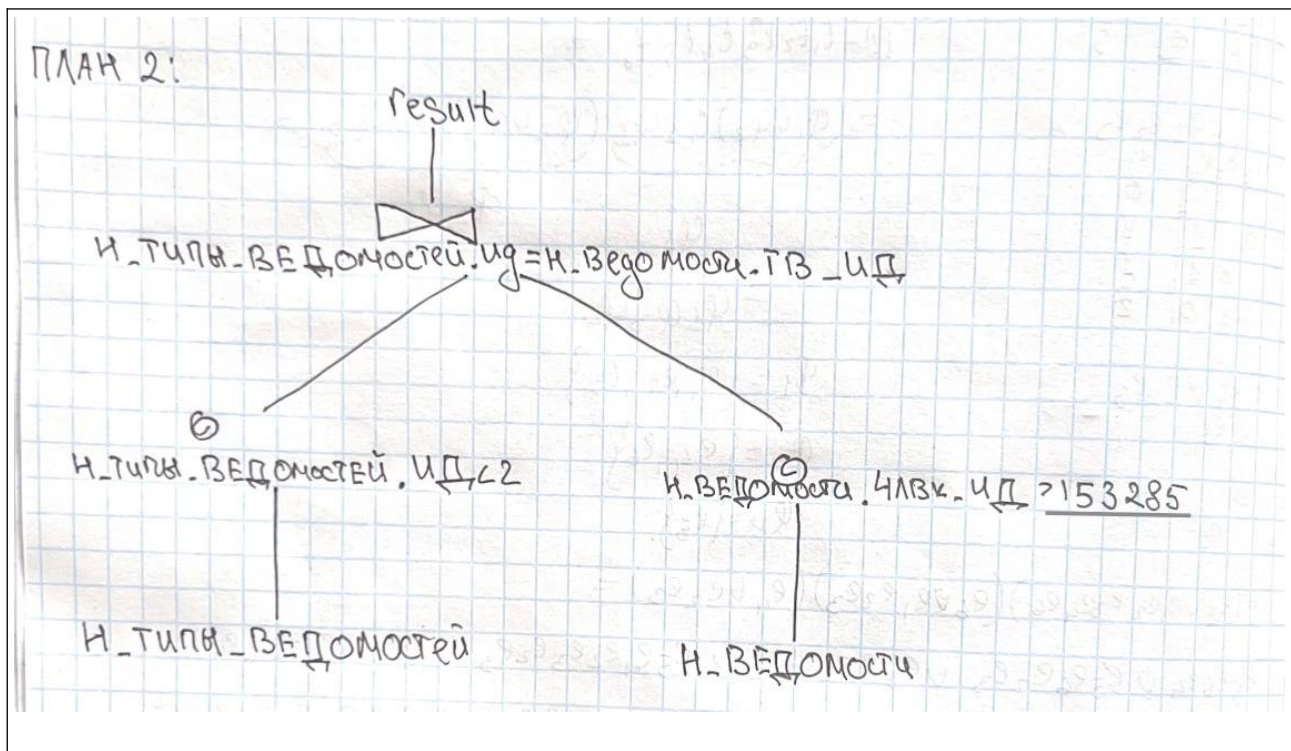
План 1:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ с применением фильтра Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2
- Полный скан таблицы Н_ВЕДОМОСТИ
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД
- Фильтрация результата соединения по условию Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285



План 2:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ с применением фильтра Н_ТИПЫ_ВЕДОМОСТЕЙ < 2
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра Н_ВЕДОМОСТИ.ЧЛВК_ИД > 153285
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД



Оптимальный план: план 2 (фильтрация данных выполняется до объединения таблиц, что уменьшает количество строк для обработки соединения)

При добавлении индексов планы выполнения:

- Вместо полного скана таблиц будет использоваться индексный скан
- Nested Loops Join станет быстрее благодаря индексу на атрибуте ТВ_ИД

4. План выполнения запроса 1

Hash Join (cost=103.20..4338.94 rows=3004 width=30)

Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")

-> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=102.15..4280.81 rows=9013 width=8)

Recheck Cond: ("ЧЛВК_ИД" > 153285)

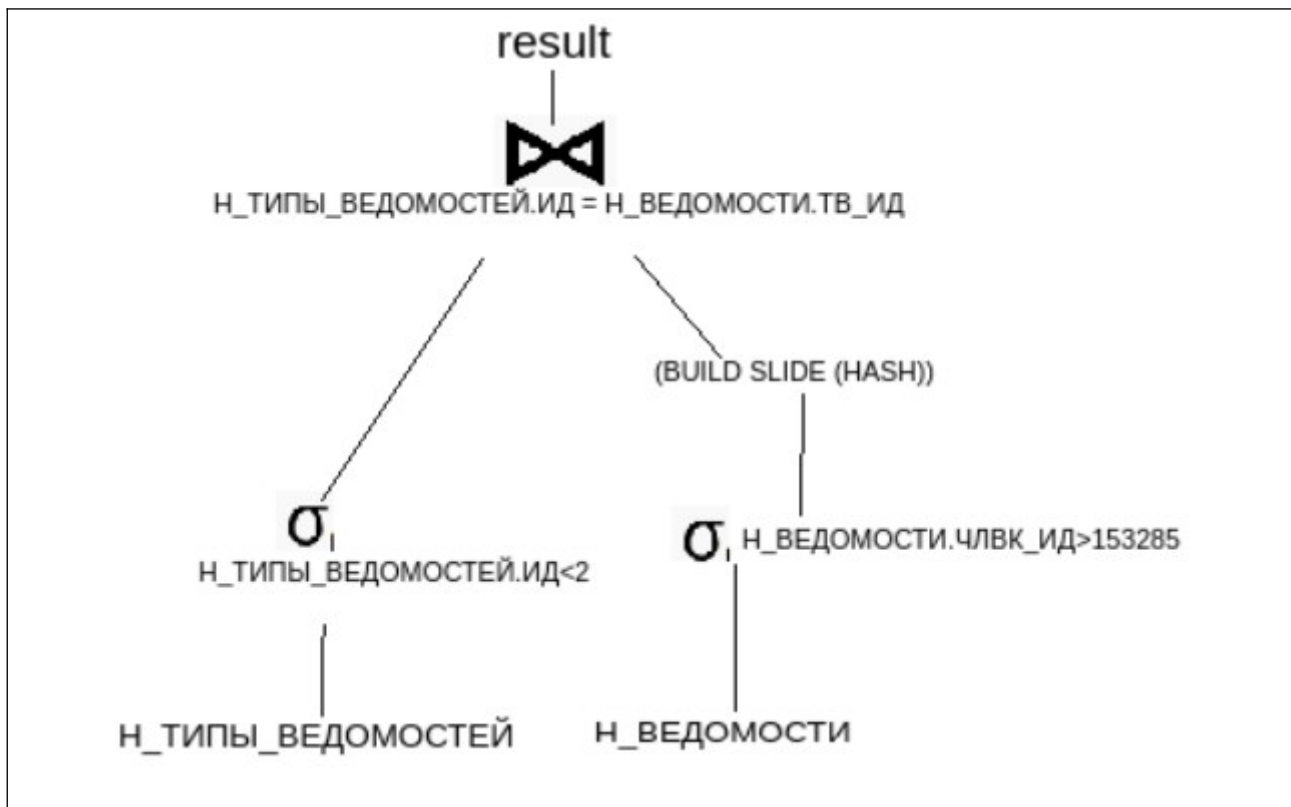
-> Bitmap Index Scan on "ВЕД_ЧЛВК_FK_IFK" (cost=0.00..99.89 rows=9013 width=0)

Index Cond: ("ЧЛВК_ИД" > 153285)

-> Hash (cost=1.04..1.04 rows=1 width=30)

-> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=30)

Filter: ("ИД" < 2)



5. Уменьшение выполнения времени 2 запроса

Индексы, добавление которых уменьшит время выполнения

1) Н_ЛЮДИ

- Индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием `Н_ЛЮДИ.ИД > 152862`

2) Н_ВЕДОМОСТИ

- Индекс на атрибуте ДАТА (B-tree). Это ускорит фильтрацию строк с условием `Н_ВЕДОМОСТИ.ДАТА < 2022-06-08`.

3) Н_СЕССИЯ

- Индекс на атрибуте ЧЛВК_ИД (B-tree). Это ускорит соединение таблиц по условию `Н_СЕССИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД`

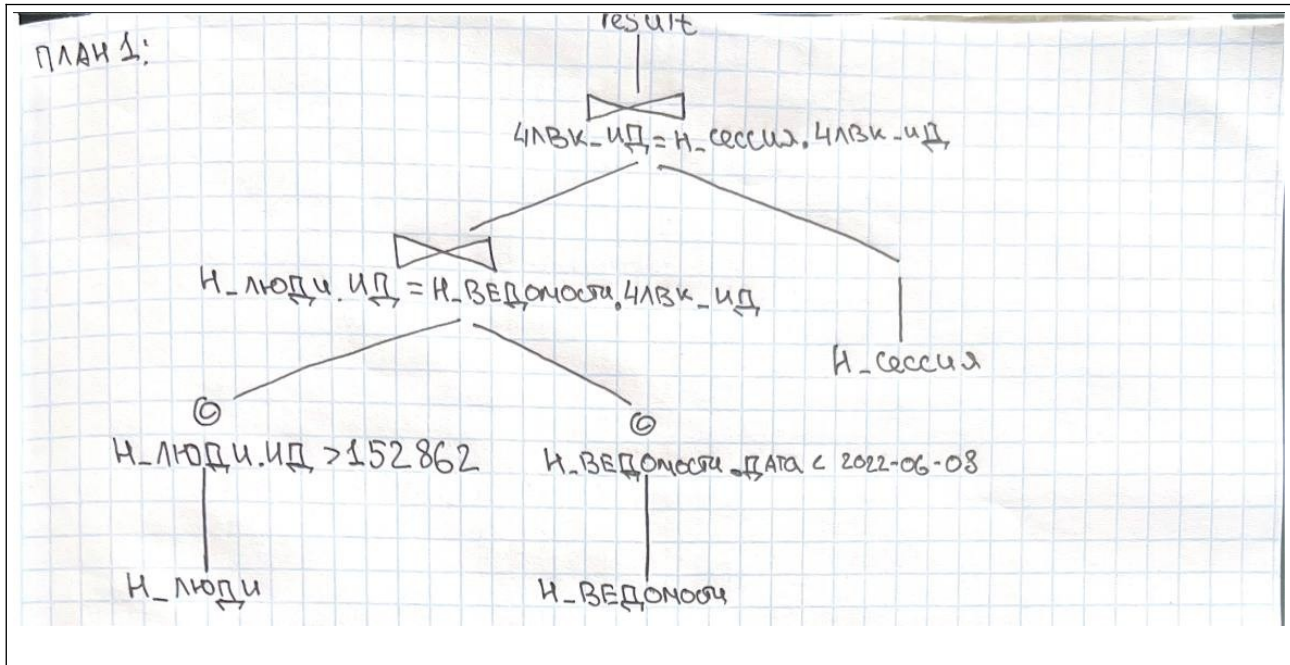
Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запросов без индексов

План 1:

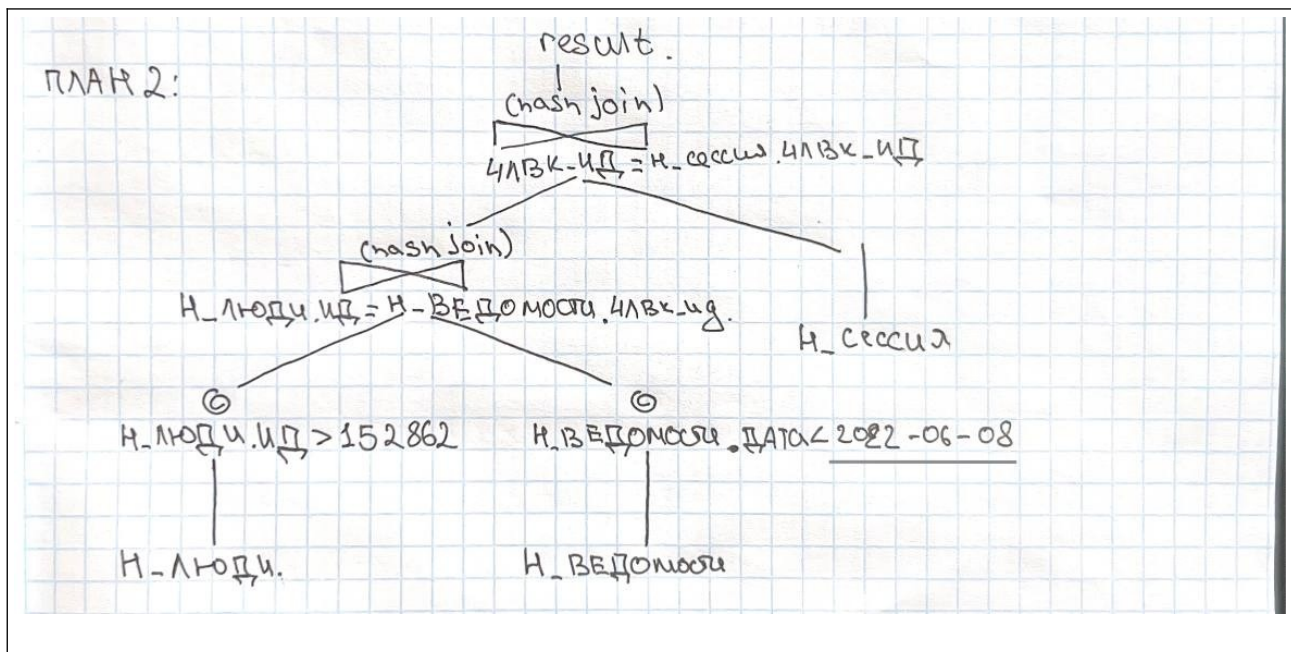
- Полный скан таблицы Н_ЛЮДИ с применением фильтра `Н_ЛЮДИ.ИД > 152862`
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра `Н_ВЕДОМОСТИ.ДАТА < 2022-06-08`

- Соединение таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД
- Полный скан таблицы Н_СЕССИЯ
- Соединение результатов предыдущего этапа с таблицей Н_СЕССИЯ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД



План 2:

- Полный скан таблицы Н_ЛЮДИ с применением фильтра Н_ЛЮДИ.ИД > 152862
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра Н_ВЕДОМОСТИ.ДАТА < 2022-06-08
- Соединение таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ с использованием Hash Join по атрибуту ЧЛВК_ИД
- Полный скан таблицы Н_СЕССИЯ
- Соединение результатов предыдущего этапа с таблицей Н_СЕССИЯ с использованием Hash Join по атрибуту ЧЛВК_ИД



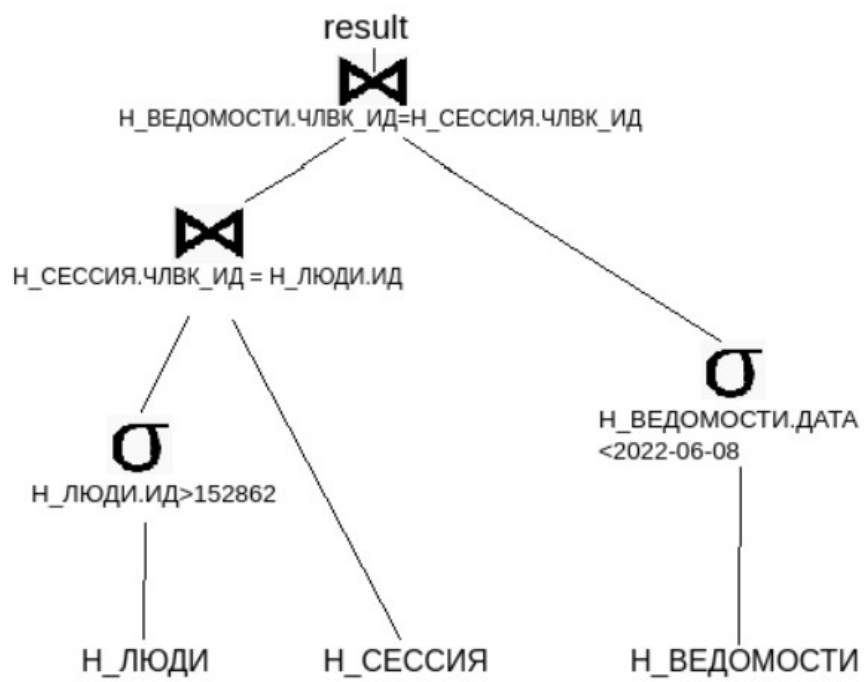
Оптимальный план: План 1, так как в случае небольшого набора обрабатываемых данных и заранее ожидаемых малых результатов фильтрации, Nested Loops Join предпочтительнее Hash Join`у.

При добавлении индексов планы выполнения запросов изменятся:

- Вместо полного скана таблиц будет использоваться индексный скан
- Nested Loops Join станет быстрее благодаря индексам на атрибутах ЧЛВК_ИД

6. План выполнения запроса 2

```
Nested Loop (cost=0.60..2003.67 rows=22520 width=21)
  Join Filter: ("Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
  -> Nested Loop (cost=0.29..357.52 rows=518 width=21)
    -> Seq Scan on "Н_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=4)
    -> Memoize (cost=0.29..0.87 rows=1 width=17)
      Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
      Cache Mode: logical
    -> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..0.86 rows=1 width=17)
      Index Cond: (("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ИД" > 152862))
  -> Memoize (cost=0.30..6.16 rows=68 width=8)
    Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
    Cache Mode: logical
    -> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..6.15
rows=68 width=8)
      Index Cond: ("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
      Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)
```



7.Вывод

Во время выполнения лабораторной работы я научился строить планы выполнения запросов, изучил различные виды индексов и методы соединения. Воспользовался полученными знаниями для планирования оптимизации запросов.