МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«БАЗЫ ДАННЫХ»

Вариант № **283772**

***Выполнил:***Рязанов Никита Сергеевич

***Проверила:***Байрамова Хумай Бахруз Кызы

**Содержание**

[Задание 3](#_Toc198133510)

[Реализация и анализ запросов 4](#_Toc198133511)

[Заключение 10](#_Toc198133512)

Задание

Составить запросы на языке **SQL** (пункты 1–2).

Для каждого запроса предложить индексы, добавление которых уменьшит время выполнения запроса.

Для запросов необходимо составить возможные планы их выполнения. 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 необходимо добавить в отчет вывод команды **EXPLAIN ANALYZE**.

Подробные ответы на все вышеперечисленные вопросы должны присутствовать в отчете.

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.ЧЛВК\_ИД.  
   **Фильтры** (AND):
   * Н\_ОЦЕНКИ.КОД < зачет.
   * Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

**Вид соединения**: RIGHT JOIN.

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   **Таблицы**: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
   Вывести **атрибуты**: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ДАТА.  
   **Фильтры** (AND):
   * Н\_ЛЮДИ.ИМЯ = Владимир.
   * Н\_ВЕДОМОСТИ.ДАТА < 1998-01-05.
   * Н\_СЕССИЯ.ДАТА > 2004-01-17.

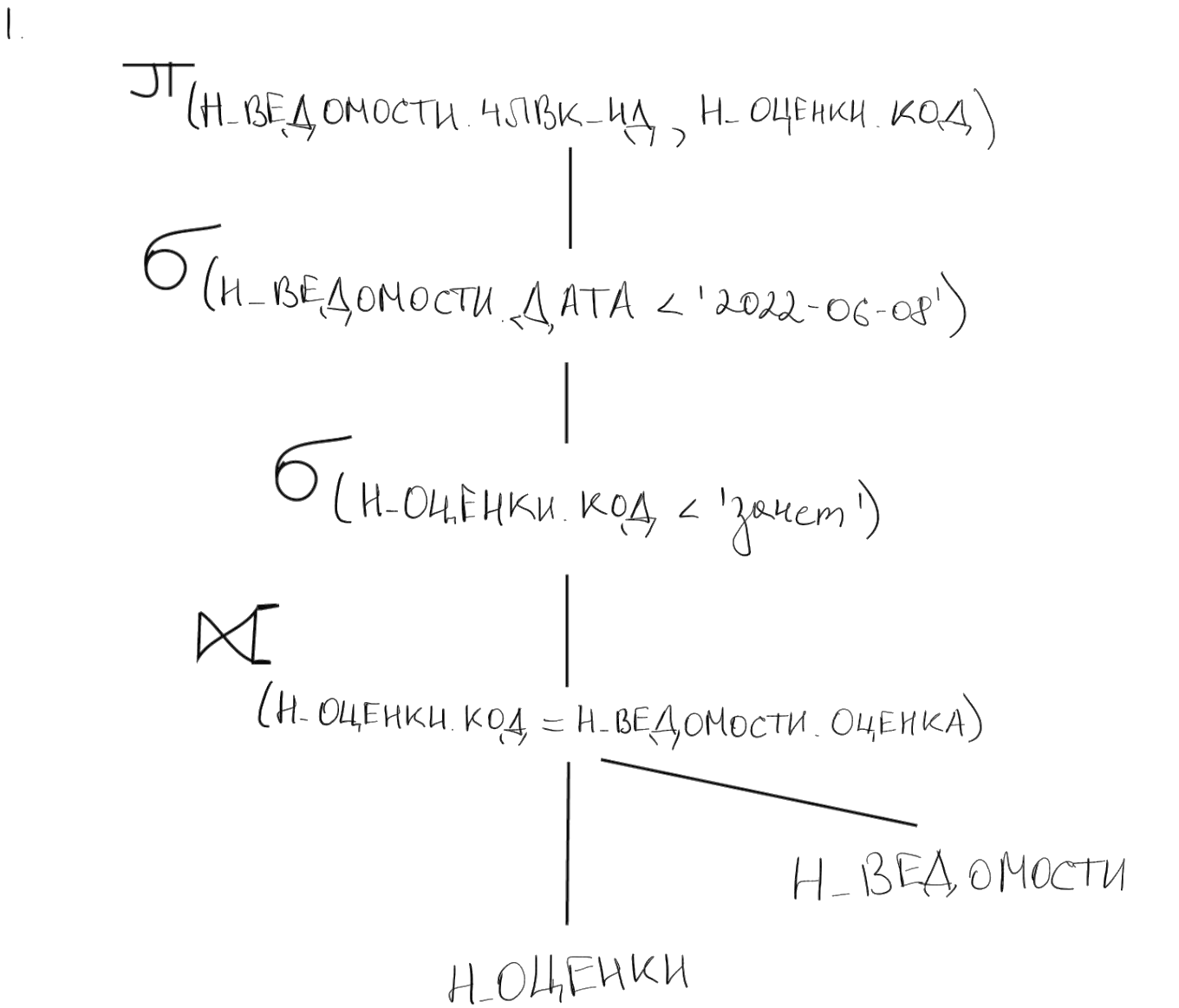
**Вид соединения**: RIGHT JOIN.

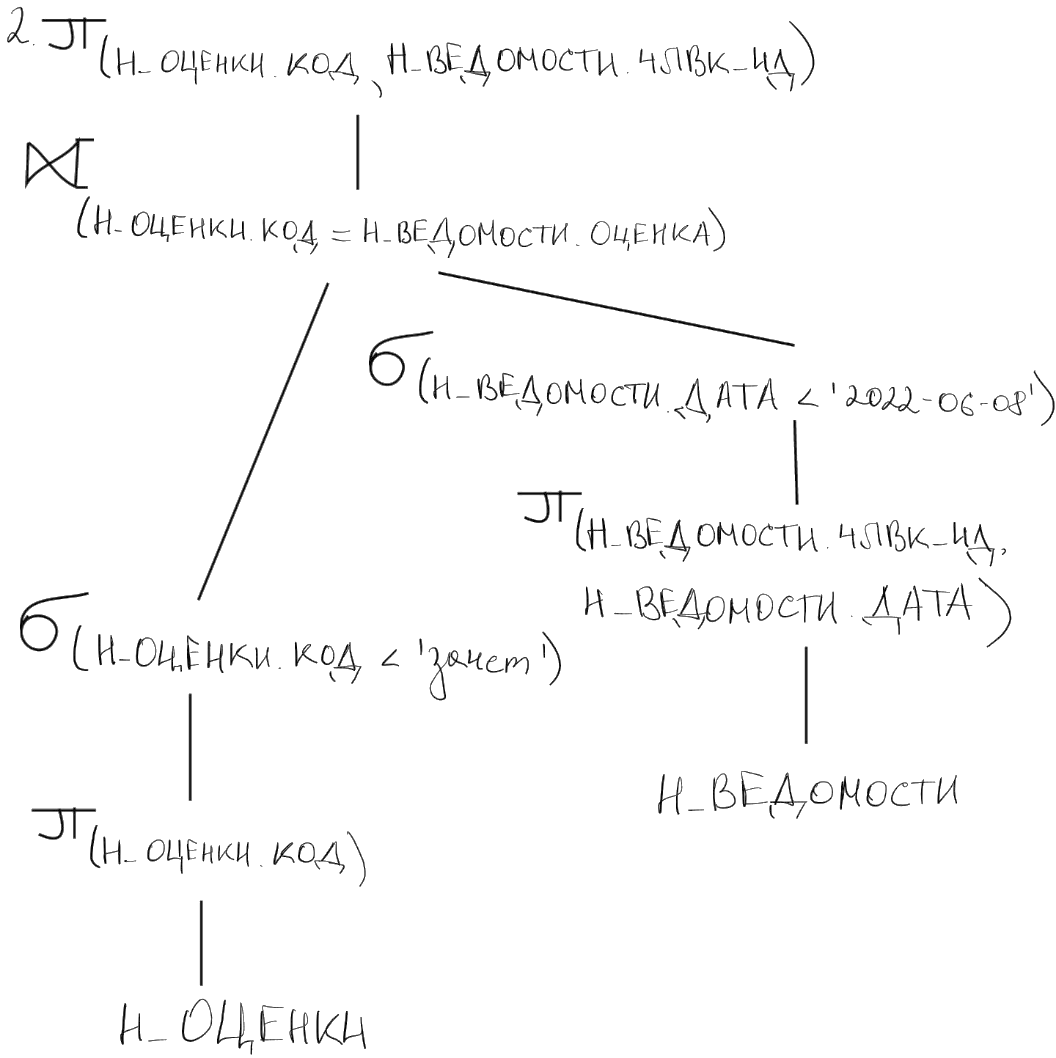
Реализация и анализ запросов

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.ЧЛВК\_ИД.  
   **Фильтры** (AND):
   * Н\_ОЦЕНКИ.КОД < зачет.
   * Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

**Вид соединения**: RIGHT JOIN.

|  |
| --- |
| SELECT  "Н\_ОЦЕНКИ"."КОД",  "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  FROM "Н\_ОЦЕНКИ"  RIGHT JOIN "Н\_ВЕДОМОСТИ" ON ("Н\_ОЦЕНКИ"."КОД" = "Н\_ВЕДОМОСТИ"."ОЦЕНКА")  WHERE ("Н\_ОЦЕНКИ"."КОД" < 'зачет'  AND "Н\_ВЕДОМОСТИ"."ДАТА" < '2022-06-08'); |

Возможные **планы выполнения** запроса:



Из представленных планов оптимальным является 2-й, так как сначала мы отфильтровываем нужные нам значения атрибутов, и только затем делаем объединение, в то время как в 1-м плане мы сначала соединяем таблицы целиком и затем берем нужные значения.

**Индексы**:

|  |
| --- |
| CREATE INDEX "ИНДЕКС\_Н\_ВЕДОМОСТИ\_ОЦЕНКА" ON "Н\_ВЕДОМОСТИ" USING hash("ОЦЕНКА");  CREATE INDEX "ИНДЕКС\_Н\_ВЕДОМОСТИ\_ДАТА" ON "Н\_ВЕДОМОСТИ" USING btree("ДАТА"); |

Добавив вышеуказанные индексы, скорость выполнения запроса должна увеличиться, так как при помощи хэш-индекса прямое сравнение будет производиться быстрее («Н\_ОЦЕНКИ».«КОД» = «Н\_ВЕДОМОСТИ».«ОЦЕНКА»), а с помощью сбалансированного дерева уменьшатся траты на операторы сравнения помимо прямого (в данном случае для «Н\_ВЕДОМОСТИ».«ДАТА» < ‘2022-06-08’).

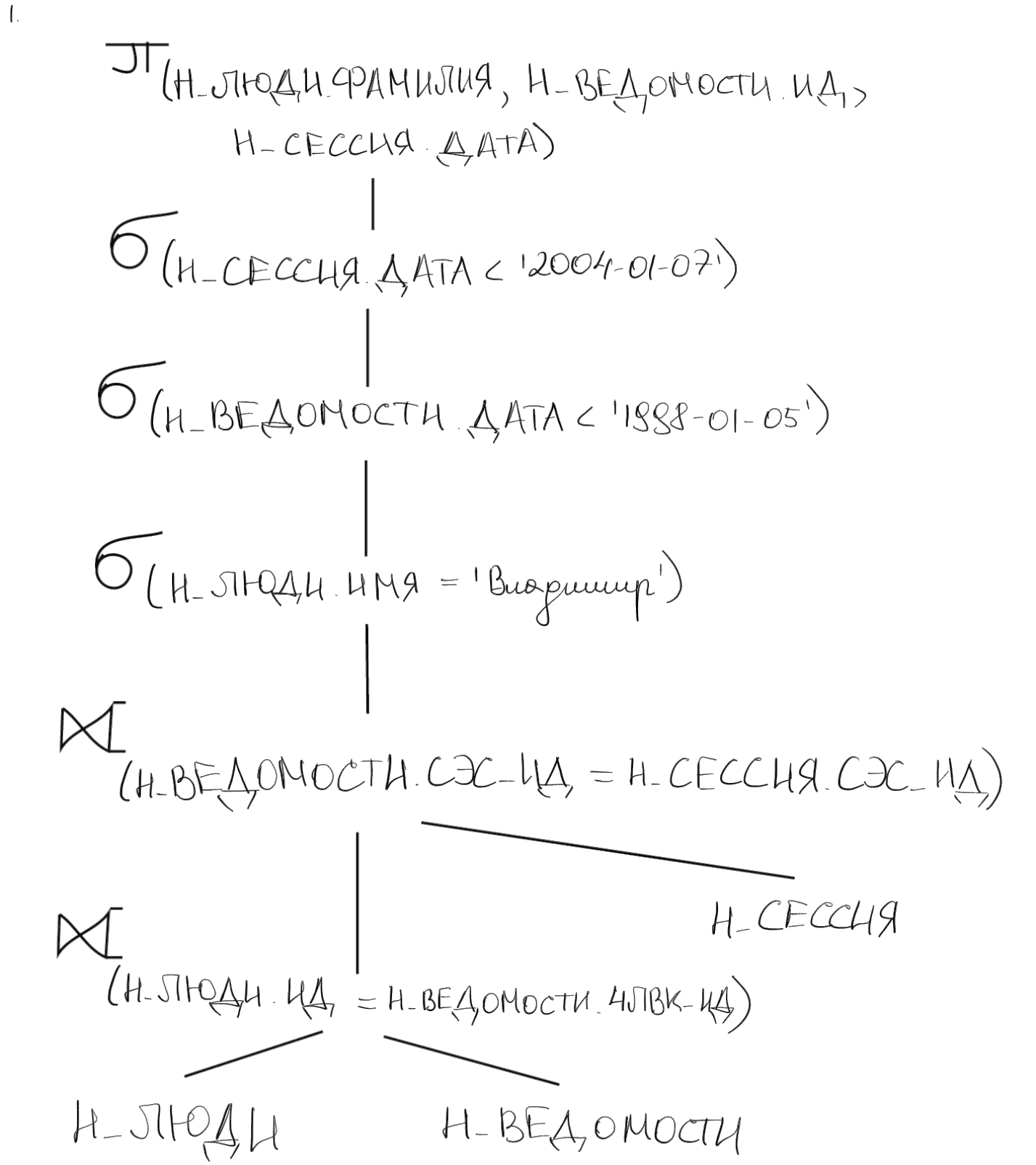
Вывод команды **EXPLAIN ANALYSE** для данного запроса:

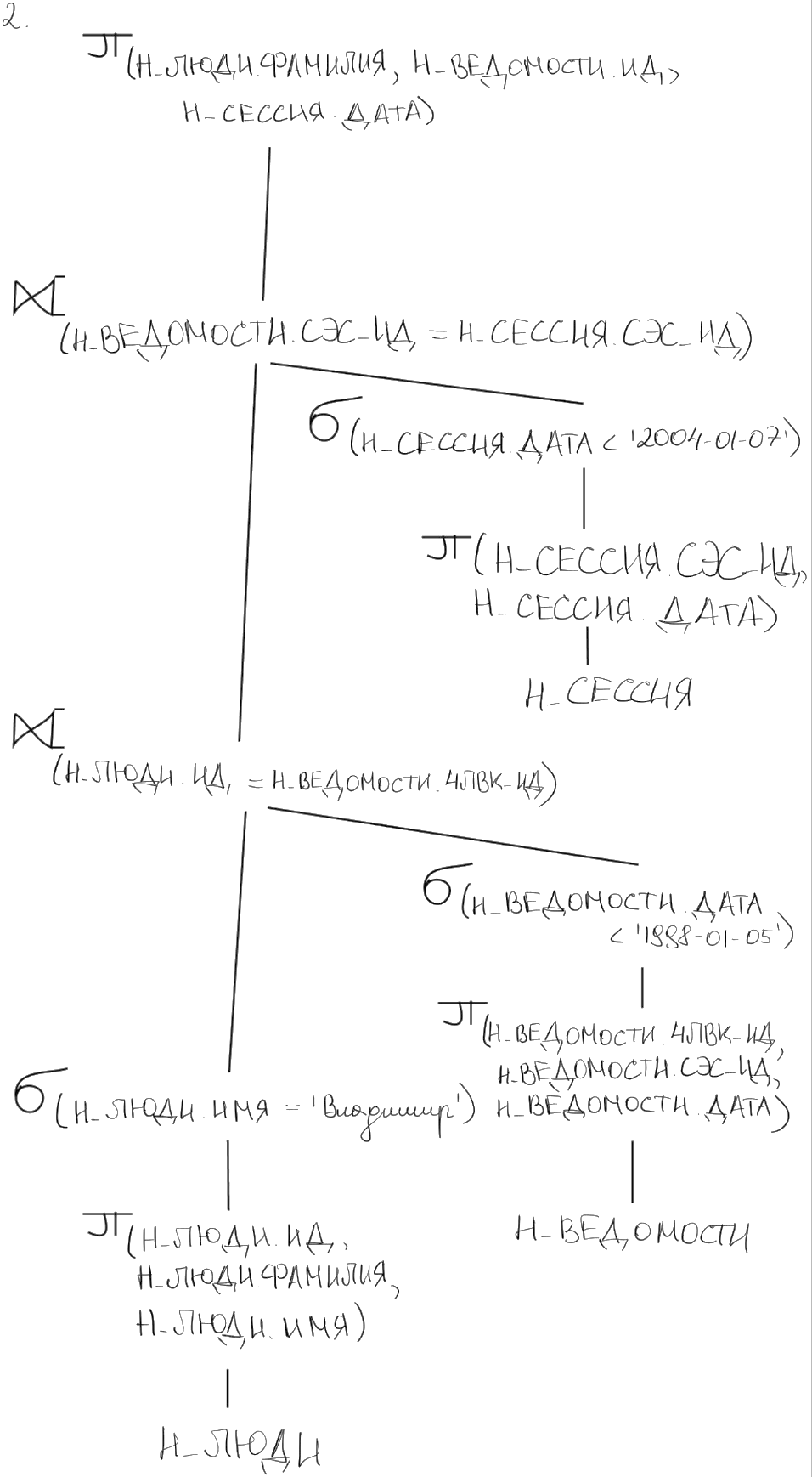
|  |
| --- |
| Hash Join (cost=1.15..7744.23 rows=74122 width=38) (actual time=0.062..75.808 rows=113487 loops=1)  Hash Cond: (("Н\_ВЕДОМОСТИ"."ОЦЕНКА")::text = ("Н\_ОЦЕНКИ"."КОД")::text)  -> Seq Scan on "Н\_ВЕДОМОСТИ" (cost=0.00..6884.50 rows=222367 width=10) (actual time=0.007..37.531 rows=222437 loops=1)  Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)  Rows Removed by Filter: 3  -> Hash (cost=1.11..1.11 rows=3 width=34) (actual time=0.031..0.041 rows=5 loops=1)  Buckets: 1024 Batches: 1 Memory Usage: 9kB  -> Seq Scan on "Н\_ОЦЕНКИ" (cost=0.00..1.11 rows=3 width=34) (actual time=0.021..0.023 rows=5 loops=1)  Filter: (("КОД")::text < 'зачет'::text)  Rows Removed by Filter: 4  Planning Time: 0.984 ms  Execution Time: 80.979 ms |

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   **Таблицы**: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
   Вывести **атрибуты**: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ДАТА.  
   **Фильтры** (AND):
   * Н\_ЛЮДИ.ИМЯ = Владимир.
   * Н\_ВЕДОМОСТИ.ДАТА < 1998-01-05.
   * Н\_СЕССИЯ.ДАТА > 2004-01-17.

**Вид соединения**: RIGHT JOIN.

|  |
| --- |
| SELECT  "Н\_ЛЮДИ"."ФАМИЛИЯ",  "Н\_ВЕДОМОСТИ"."ИД",  "Н\_СЕССИЯ"."ДАТА"  FROM "Н\_ЛЮДИ"  RIGHT JOIN "Н\_ВЕДОМОСТИ" ON ("Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  RIGHT JOIN "Н\_СЕССИЯ" ON ("Н\_ВЕДОМОСТИ"."СЭС\_ИД" = "Н\_СЕССИЯ"."СЭС\_ИД")  WHERE ("Н\_ЛЮДИ"."ИМЯ" = 'Владимир'  AND "Н\_ВЕДОМОСТИ"."ДАТА" < '1998-01-05'  AND "Н\_СЕССИЯ"."ДАТА" > '2004-01-17'); |

Возможные **планы выполнения** запроса:



Из представленных планов оптимальным является 2-й, так как сначала мы отфильтровываем нужные нам значения атрибутов с помощью выборки и проекции, и только затем делаем объединение, а в 1-м плане мы сначала соединяем таблицы целиком и только затем берем нужные значения, что явно не является лучшим решением.

**Индексы**:

|  |
| --- |
| CREATE INDEX "ИНДЕКС\_Н\_ЛЮДИ\_ИМЯ" ON "Н\_ЛЮДИ" USING hash("ИМЯ");  CREATE INDEX "ИНДЕКС\_Н\_ВЕДОМОСТИ\_ДАТА" ON "Н\_ВЕДОМОСТИ" USING btree("ДАТА");  CREATE INDEX "ИНДЕКС\_Н\_СЕССИЯ\_ДАТА" ON "Н\_СЕССИЯ" USING btree("ДАТА");  CREATE INDEX "ИНДЕКС\_Н\_ВЕДОМОСТИ\_ЧЛВК\_ИД" ON "Н\_ВЕДОМОСТИ" USING hash("ЧЛВК\_ИД");  CREATE INDEX "ИНДЕКС\_Н\_СЕССИЯ\_СЭС\_ИД" ON "Н\_СЕССИЯ" USING hash("СЭС\_ИД"); |

Добавив вышеуказанные индексы, скорость выполнения запроса должна увеличиться, так как при помощи хэш-индекса прямое сравнение будет производиться быстрее:

* «Н\_ЛЮДИ».«ИМЯ» = ‘Владимир’
* «Н\_ЛЮДИ».«ИД» = «Н\_ВЕДОМОСТИ».«ЧЛВК\_ИД»
* «Н\_ВЕДОМОСТИ».«СЭС\_ИД» = «Н\_СЕССИЯ».«СЭС\_ИД»

А с помощью сбалансированного дерева уменьшатся траты на операторы сравнения помимо прямого, в данном случае для:

* «Н\_ВЕДОМОСТИ».«ДАТА» < ‘1998-01-05’
* «Н\_СЕССИЯ».«ДАТА» < ‘2004-01-07’

Вывод команды **EXPLAIN ANALYSE** для данного запроса:

|  |
| --- |
| Nested Loop (cost=0.86..25.92 rows=1 width=28) (actual time=0.006..0.007 rows=0 loops=1)  -> Nested Loop (cost=0.58..15.63 rows=1 width=24) (actual time=0.005..0.006 rows=0 loops=1)  -> Index Scan using "ВЕД\_ДАТА\_I" on "Н\_ВЕДОМОСТИ" (cost=0.29..7.19 rows=1 width=12) (actual time=0.005..0.005 rows=0 loops=1)  Index Cond: ("ДАТА" < '1998-01-05 00:00:00'::timestamp without time zone)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=20) (never executed)  Index Cond: ("ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  Filter: (("ИМЯ")::text = 'Владимир'::text)  -> Index Scan using "СЕС\_СЭС\_FK" on "Н\_СЕССИЯ" (cost=0.28..10.27 rows=1 width=12) (never executed)  Index Cond: ("СЭС\_ИД" = "Н\_ВЕДОМОСТИ"."СЭС\_ИД")  Filter: ("ДАТА" > '2004-01-17 00:00:00'::timestamp without time zone)  Planning Time: 0.583 ms  Execution Time: 0.057 ms |

Заключение

В ходе проделанной лабораторной работы было изучены способы оптимизации выполнения запросов. Кроме того, были составлены возможные планы выполнения запросов, для которых были предложены индексы, способствующие ускорению их выполнения.