

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Программная инженерия
Дисциплина «Информационные системы и базы данных»

Лабораторная работа №4 *Вариант 1466*

Студент
Армут Алина
P3114

Преподаватель:
Горбунов Михаил

Санкт-Петербург, 2023 г.

Please, enter your variant: 1466

Внимание! У разных вариантов разный 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. 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:
Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ЧЛВК_ИД.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Экзаменационный лист.
б) Н_ВЕДОМОСТИ.ДАТА < 1998-01-05.
с) Н_ВЕДОМОСТИ.ДАТА = 2022-06-08.
Вид соединения: RIGHT 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.ИД, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД.
Фильтры (AND):
а) Н_ЛЮДИ.ИД < 100865.
б) Н_ВЕДОМОСТИ.ДАТА > 2010-06-18.
Вид соединения: INNER JOIN.

1)

```
SELECT "Н ТИПЫ ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ",  
       "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"  
FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ" "Н_ВЕДОМОСТИ"  
RIGHT JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ТВ_ИД" =  
"Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"  
WHERE "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" < 'Экзаменационный  
лист'  
AND "Н_ВЕДОМОСТИ"."ДАТА" < '1998-01-05'  
AND "Н_ВЕДОМОСТИ"."ДАТА" = '2022-06-08';
```

NASH. Поддерживает операцию “=” (но не поддерживают операции сравнения <, >) скорость выполнения $O(1)$, то есть константа и не зависит от объема данных.

Поэтому в данном случае используем тип индекса B-tree.

```
CREATE INDEX НАИМЕНОВАНИЕ_ИНДЕКС  
ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" ("НАИМЕНОВАНИЕ") ;
```

Будет эффективно использовать этот индекс так как: из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор "<" и данный атрибут будет редко изменяться, а значит, индекс не придется часто менять. Индекс позволит сократить время выполнения запроса.

```
CREATE INDEX index_date  
ON "Н_ВЕДОМОСТИ" USING HASH ("ДАТА");
```

Будет полезным добавить индекс на атрибут "ДАТА" из-за большого количества перебора (то есть 500+) при выполнении запроса. Индекс позволит сократить время выполнения запроса.

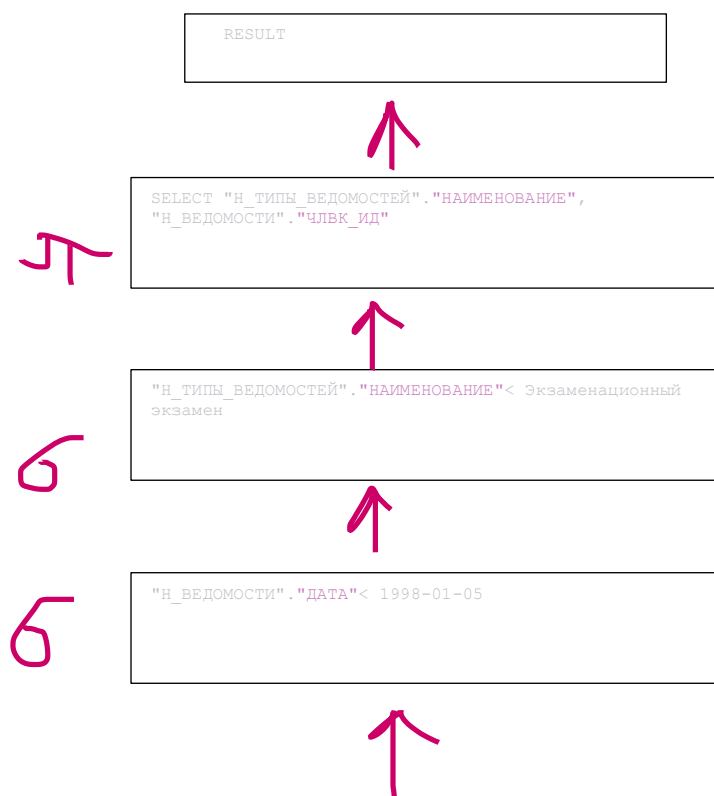
Также из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор "=".

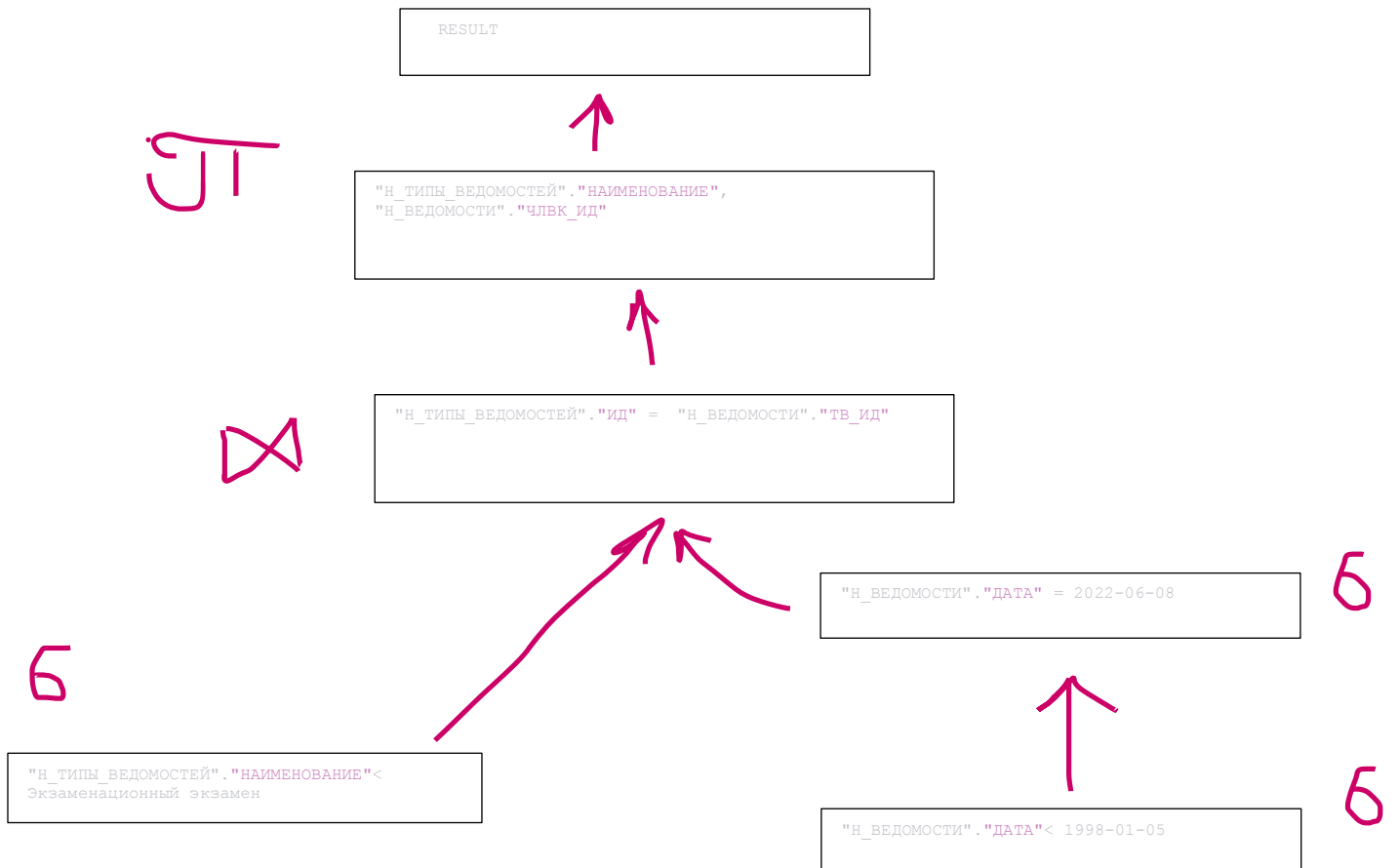
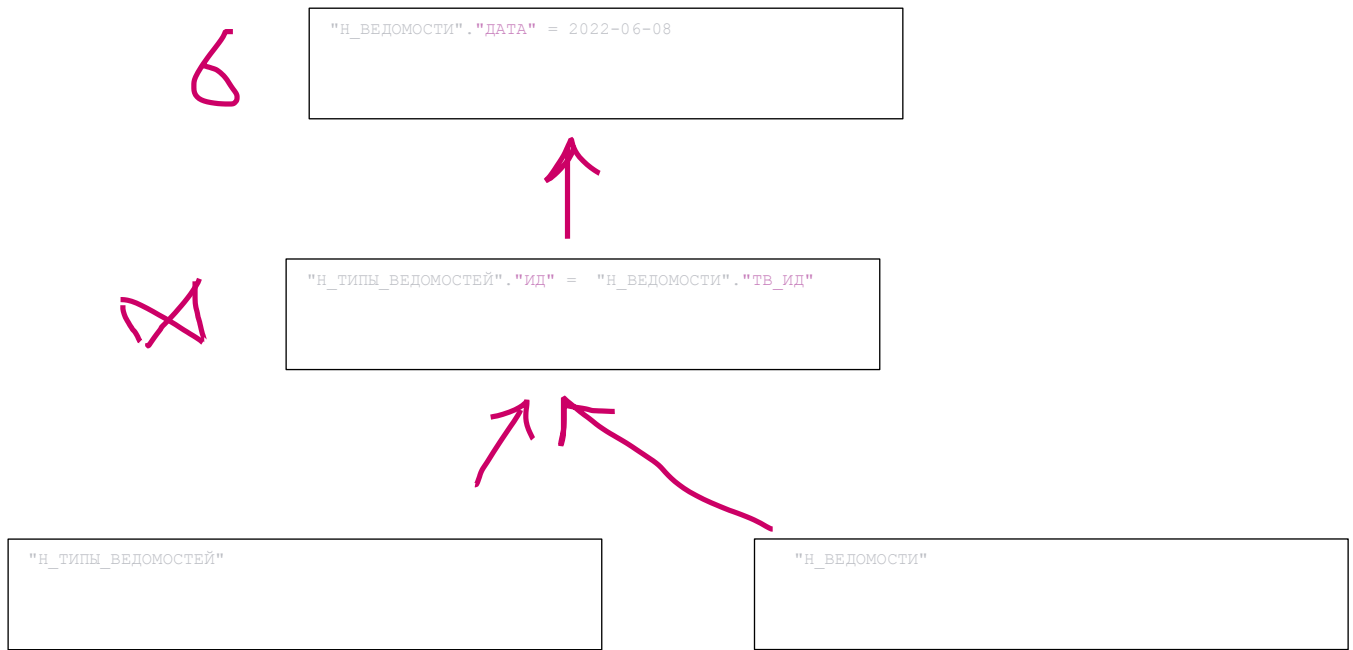
Вывод команды EXPLAIN ANALYZE:

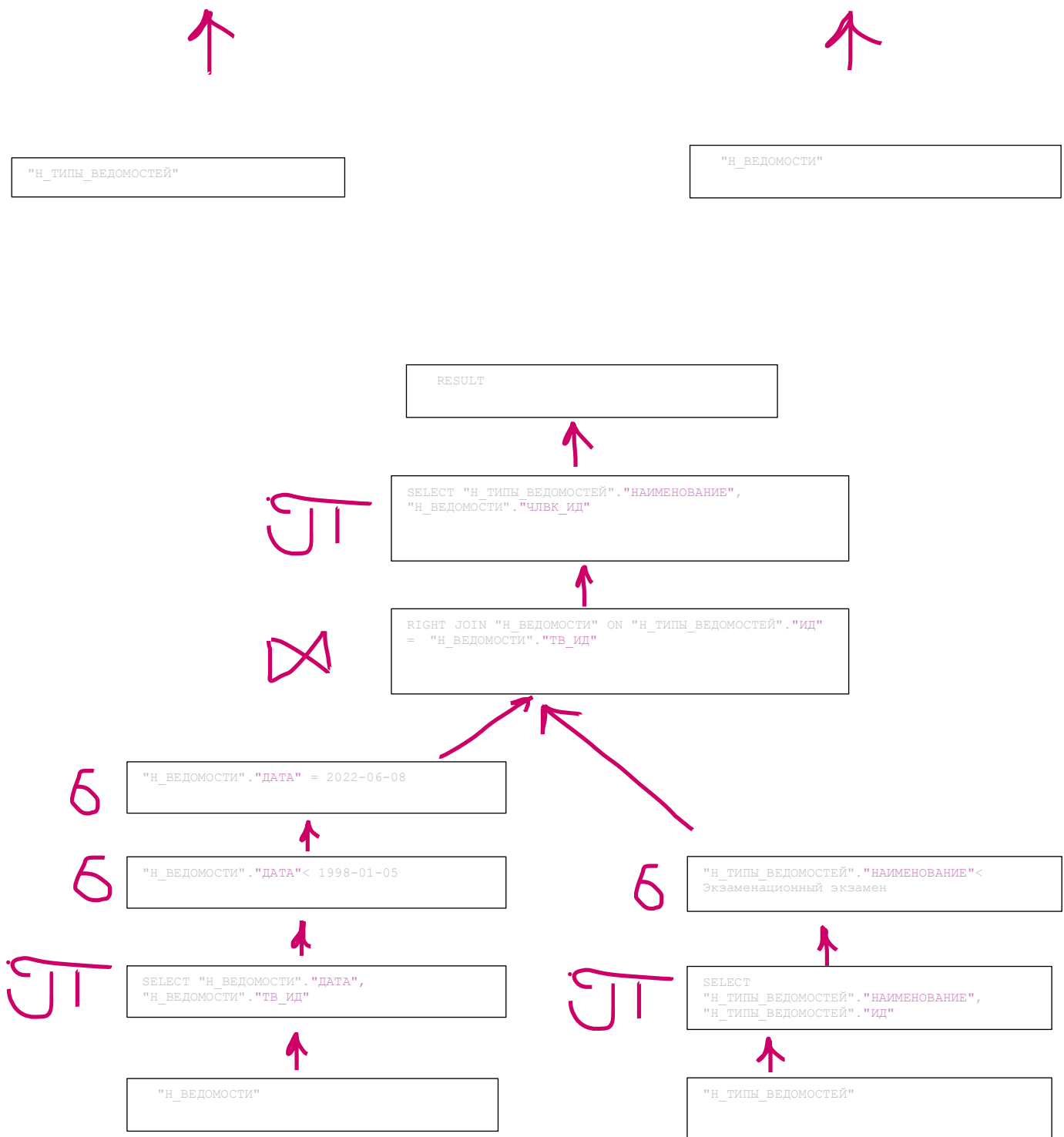
```
Nested Loop (cost=0.29..8.25 rows=1 width=422) (actual time=0.003..0.004 rows=0 loops=1)  
  Join Filter: ("Н_ВЕДОМОСТИ"."КОГДА_СОЗДАЛ" =  
    "Н_ТИПЫ_ВЕДОМОСТЕЙ"."КОГДА_СОЗДАЛ")  
    -> Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" (cost=0.29..7.20 rows=1  
      width=12) (actual time=0.002..0.002 rows=0 loops=1)  
        Index Cond: (("ДАТА" < '1998-01-05 00:00:00'::timestamp without time zone) AND  
          ("ДАТА" = '2022-06-08 00:00:00'::timestamp without time zone))  
        -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=426) (never  
          executed)  
          Filter: (("НАИМЕНОВАНИЕ")::text < 'Экзаменационный лист'::text)  
Planning Time: 1.140 ms  
Execution Time: 0.061 ms  
(8 строк)
```

План выполнения запроса:

ПЛАН 1







Более эффективным будет данный план запроса, так как в таком случае проекция сделана раньше соединения, выборка сделана раньше соединения, минимизирован размер промежуточных данных. Соединения отношение сделаны в виде правостороннего дерева.

```
SELECT "Н_ЛЮДИ"."ИД", "Н_ВЕДОМОСТИ"."ДАТА", "Н_СЕССИЯ"."УЧГОД"
FROM "Н_ЛЮДИ"
INNER JOIN "Н_ВЕДОМОСТИ" ON "Н_СЕССИЯ"."ЧЛВК_ИД" =
"Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
INNER JOIN "Н_СЕССИЯ" ON "Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE "Н_ЛЮДИ"."ИД" < 100865
AND "Н_ВЕДОМОСТИ"."ДАТА" > '2010-06-18';
```

```
CREATE INDEX index_people ON "Н_ЛЮДИ" ("ИД");
```

Будет эффективно использовать этот индекс так как: из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор "<" и данный атрибут будет редко изменяться, а значит, индекс не придется часто менять.

```
CREATE INDEX index_date ON "Н_ВЕДОМОСТИ" ("ДАТА");
```

Будет полезным добавить индекс на атрибут "ДАТА" из-за большого количества перебора (то есть 500+) при выполнении запроса. Индекс позволит сократить время выполнения запроса.

Также из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор ">".

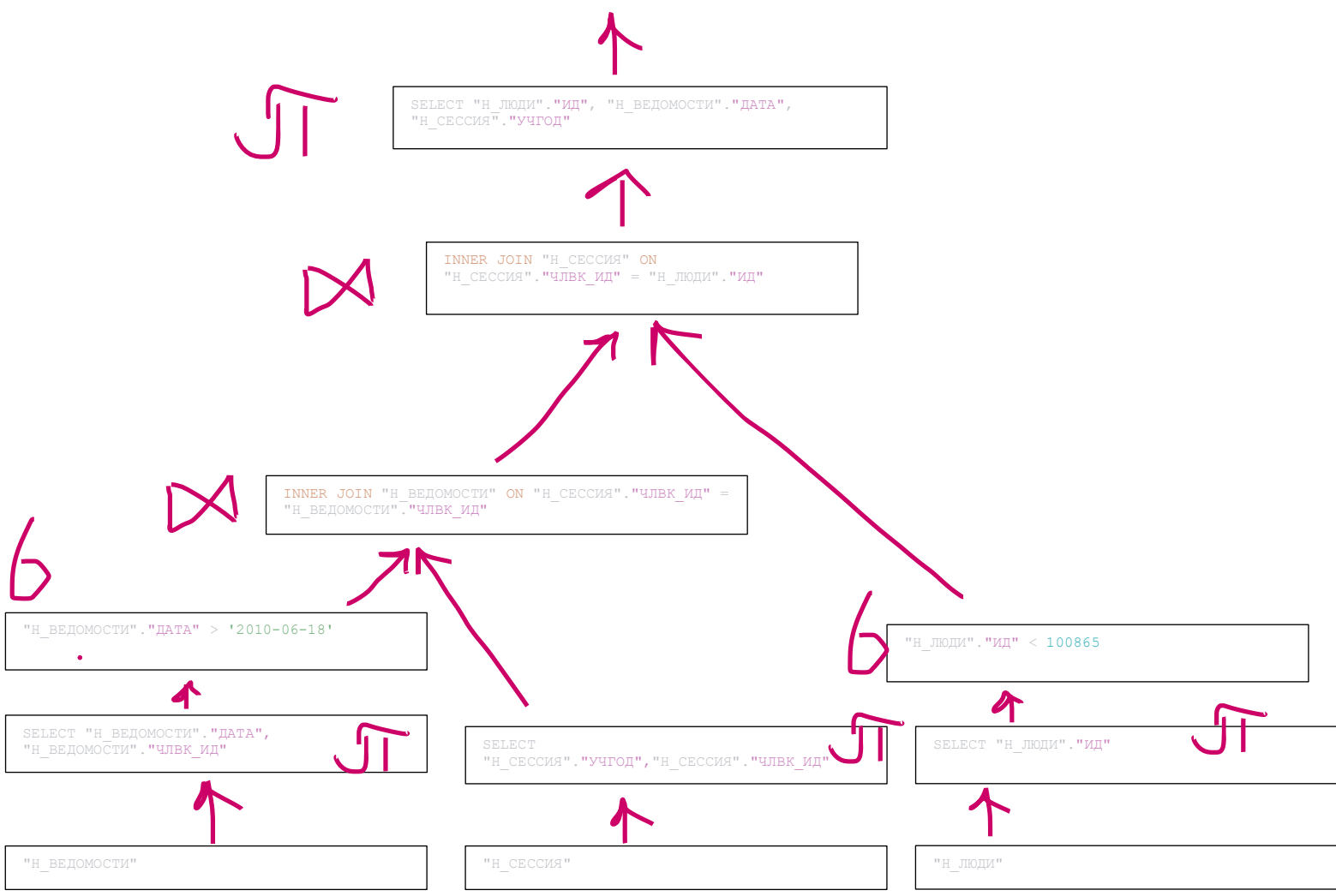
Вывод команды EXPLAIN ANALYZE:

```
Nested Loop (cost=5.36..223.16 rows=104 width=22) (actual time=3.174..3.176 rows=0
loops=1)
-> Hash Join (cost=5.06..123.44 rows=16 width=18) (actual time=0.057..1.366 rows=1641
loops=1)
    Hash Cond: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
    -> Seq Scan on "Н_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=14) (actual
time=0.008..0.425 rows=3752 loops=1)
        -> Hash (cost=4.74..4.74 rows=26 width=4) (actual time=0.022..0.023 rows=40 loops=1)
            Buckets: 1024 Batches: 1 Memory Usage: 10kB
            -> Index Only Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..4.74 rows=26
width=4) (actual time=0.004..0.009 rows=40 loops=1)
                Index Cond: ("ИД" < 100865)
                Heap Fetches: 0
        -> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..6.15 rows=8
width=12) (actual time=0.001..0.001 rows=0 loops=1641)
            Index Cond: ("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
            Filter: ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone)
    Planning Time: 1.944 ms
```

План выполнения запроса:

ПЛАН 1

RESULT



Данный план является оптимальным, так как:

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

Выводы

При выполнении лабораторной работы я познакомилась с планами выполнения запросов, индексами, командой `EXPLAIN ANALYZE`, оптимизировала запросы.