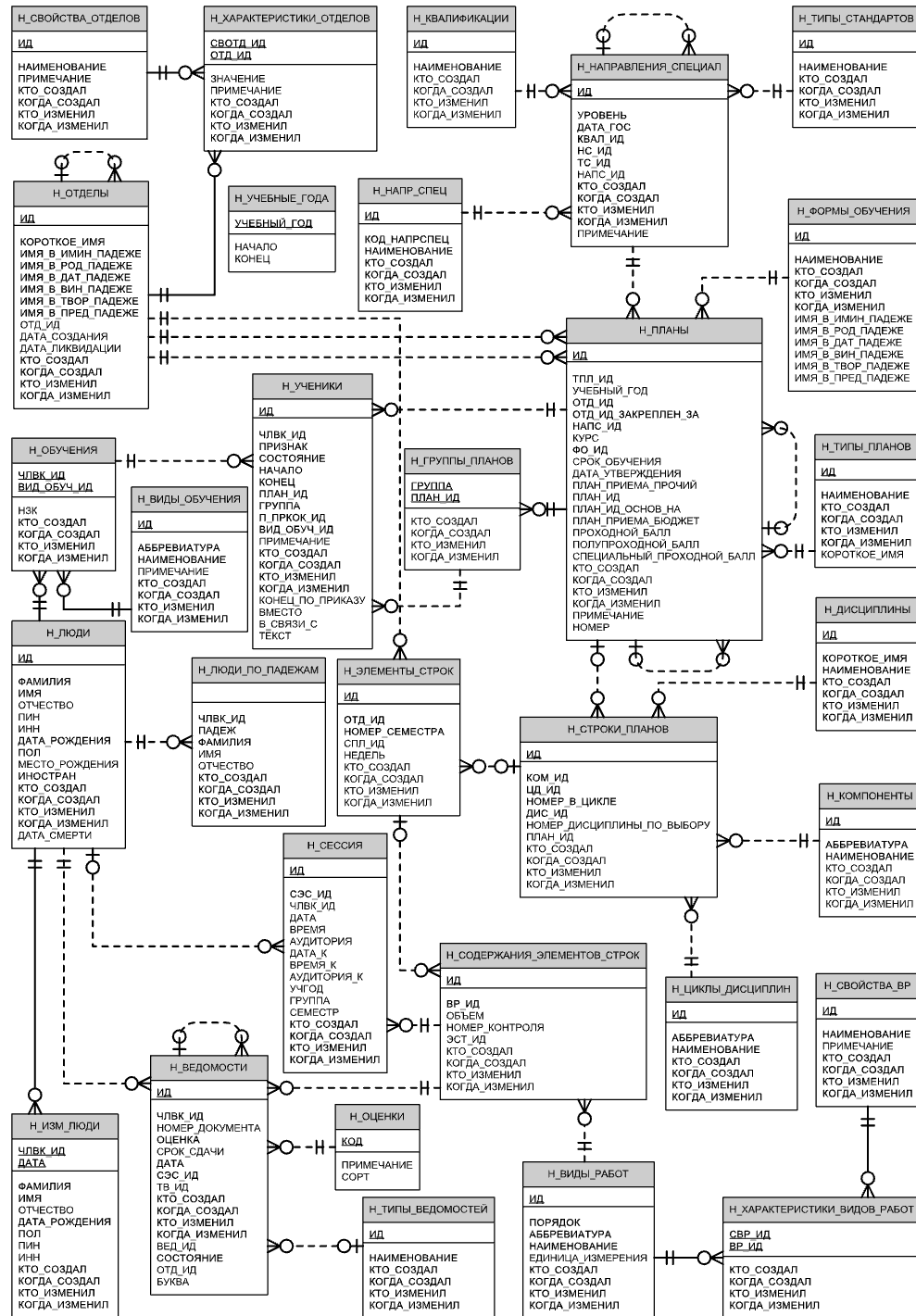


База данных



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).

Реализация запросов

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

Таблицы: *НЛЮДИ*, *НСЕССИЯ*.

Вывести атрибуты: *НЛЮДИ.ИД*, *НСЕССИЯ.УЧГОД*.

Фильтры (AND):

а) *НЛЮДИ.ИД* > 163484.

б) *НСЕССИЯ.ЧЛВК_ИД* > 100012.

Вид соединения: RIGHT JOIN.

```
SELECT Н_люди.ИД
FROM Н_люди
RIGHT JOIN
Н_сессия ON Н_люди.ИД = Н_сессия.ЧЛВК_ИД
WHERE
Н_люди.ИД > 163484 AND Н_сессия.ЧЛВК_ИД > 100012;
```

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

Таблицы: *НЛЮДИ*, *НОБУЧЕНИЯ*, *НУЧЕНИКИ*.

Вывести атрибуты: *НЛЮДИ.ИД*, *НОБУЧЕНИЯ.НЗК*, *НУЧЕНИКИ.ГРУППА*.

Фильтры: (AND)

а) *НЛЮДИ.ИД* < 100865.

б) *НОБУЧЕНИЯ.ЧЛВКИД* < 163484.

с) НУЧЕНИКИ.ИД < 39.

Вид соединения: INNER JOIN.

```
SELECT Н_люди.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА
FROM Н_люди
INNER JOIN
Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_люди.ИД
INNER JOIN
Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
WHERE
Н_люди.ИД < 100865 AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163484 AND Н_УЧЕНИКИ.ИД < 39;
```

Индексы

1 запрос

```
CREATE INDEX idx_сессия_члвк ON Н_СЕССИЯ(ЧЛВК_ИД);
CREATE INDEX idx_люди_ид ON Н_ЛЮДИ(ИД);
```

1. idxсессиячлвк выгоден, потому что по значениям НСЕССИЯ(ЧЛВКИД) происходит фильтрация, а также соединение таблиц
Индекс b-tree на атрибуте ЧЛВКИД уменьшит время выполнения запроса, потому что мы используем фильтр ЧЛВКИД > 100012, соединяем таблицы по атрибутам НЛЮДИ.ИД = НСЕССИЯ.ЧЛВК_ИД.
Hash не подходит, т.к. не поддерживает >, только точное сравнение.
GIN, GiST, SP-GiST предназначены для JSON, массивов, геоданных и прочего
2. idxлюдиид также полезен из-за фильтрации, а также соединения
Индекс b-tree на атрибуте ИД уменьшит время выполнения запроса, потому что используется фильтрация ИД >, происходит соединение ИД = ЧЛВК_ИД
Hash не удовлетворяет, так как не поддерживает сравнения >
GIN, GiST, SP-GiST — не применимы.

2 запрос

```
CREATE INDEX idx_люди_ид ON Н_ЛЮДИ(ИД);
CREATE INDEX idx_обучения_члвк ON Н_ОБУЧЕНИЯ(ЧЛВК_ИД);
CREATE INDEX idx_ученики_ид ON Н_УЧЕНИКИ(ИД);
```

1. Был выбран индекс btree, так как участвует в фильтрации: НЛЮДИ.ИД < 100865 и в соединении: НОБУЧЕНИЯ.ЧЛВКИД = НЛЮДИ.ИД

Hash не поддерживает <

GIN, GiST, SP-GiST не применимы к числам.

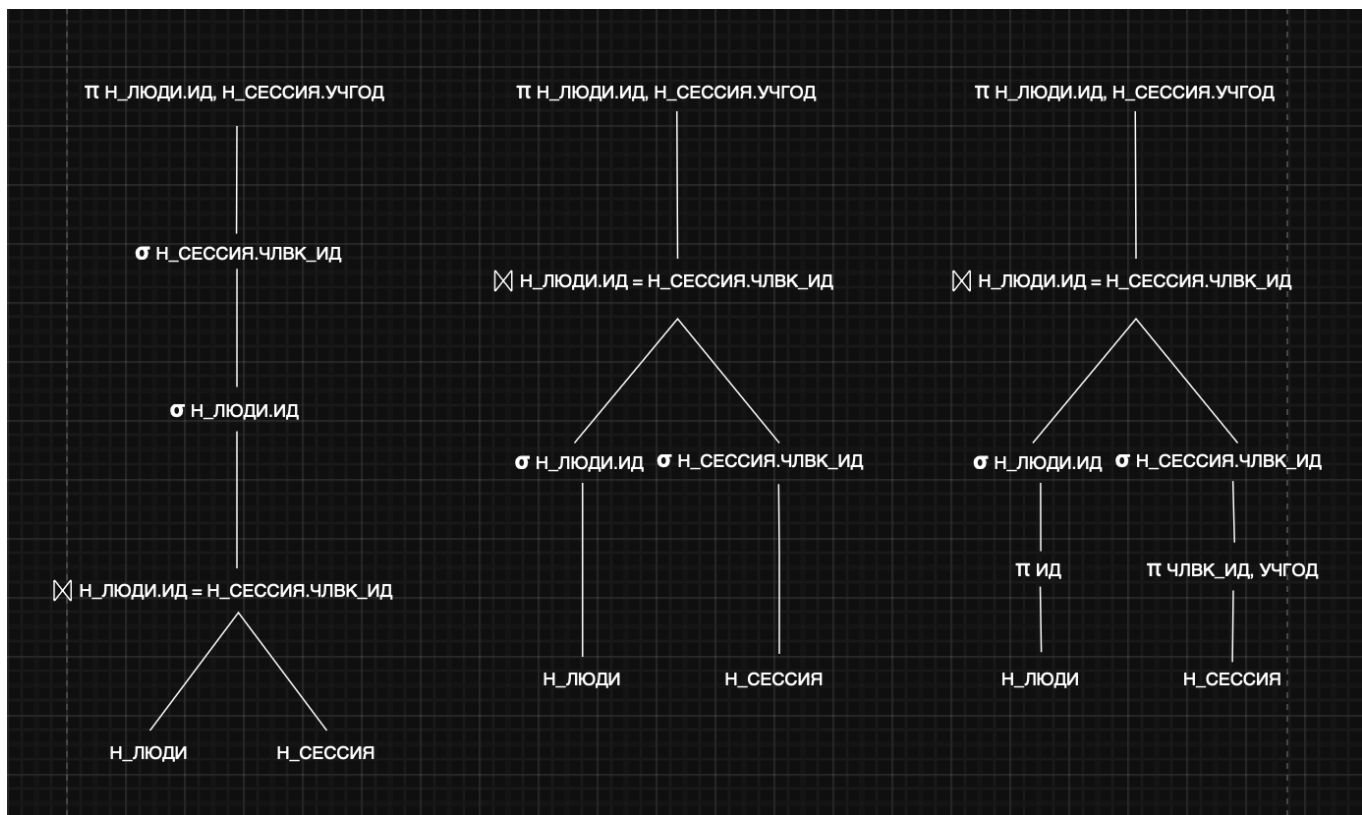
2. Индекс btree отлично подходит, так как происходит фильтрация: ЧЛВКИД < 163484, два соединения с НЛЮДИ и с Н_УЧЕНИКИ
GIN, GiST, SP-GiST не применимы к числам

Таблица Н_УЧЕНИКИ

```
CREATE INDEX idxученикиид ON Н_УЧЕНИКИ(ИД);
```

Планы

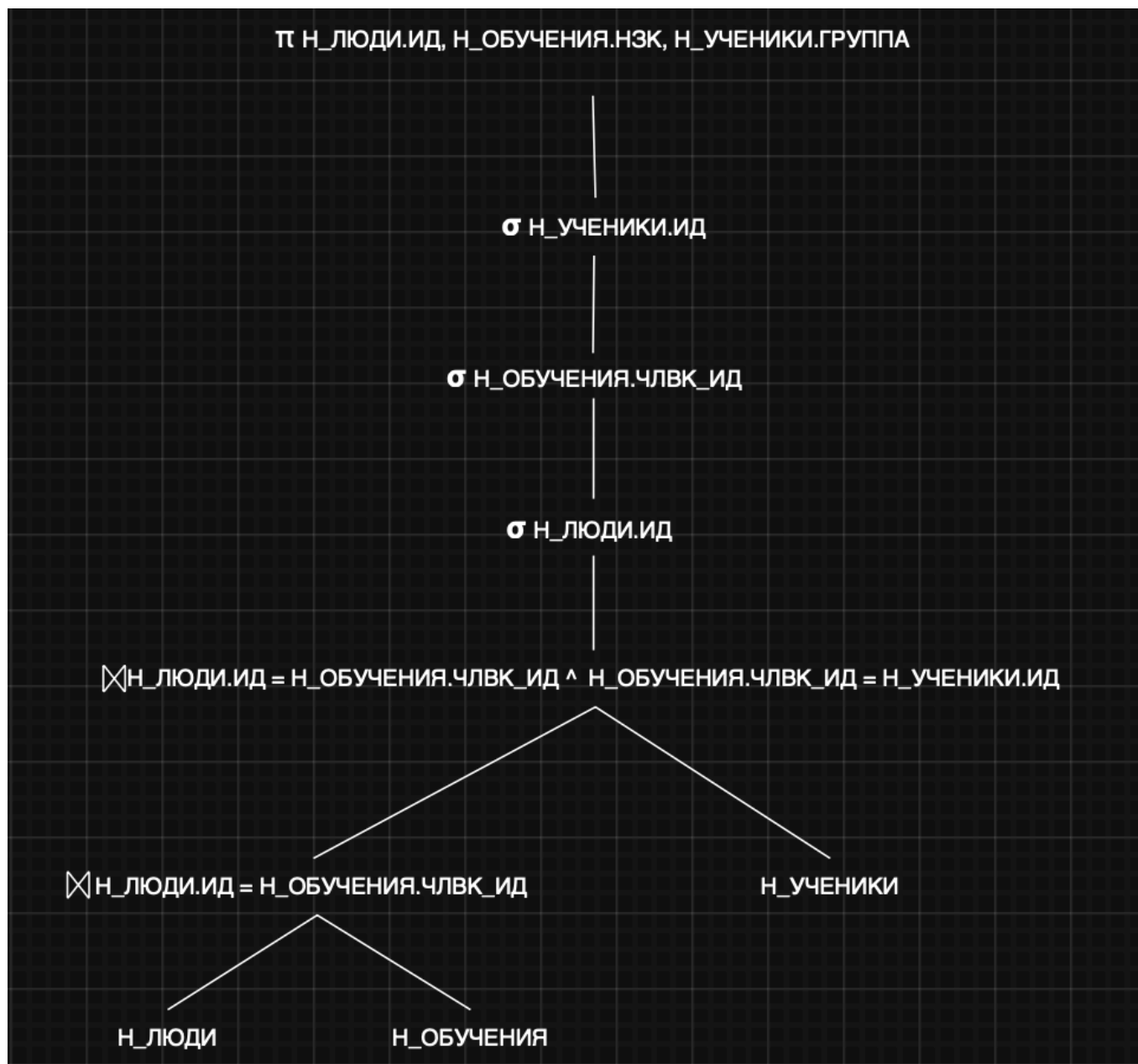
1 запрос



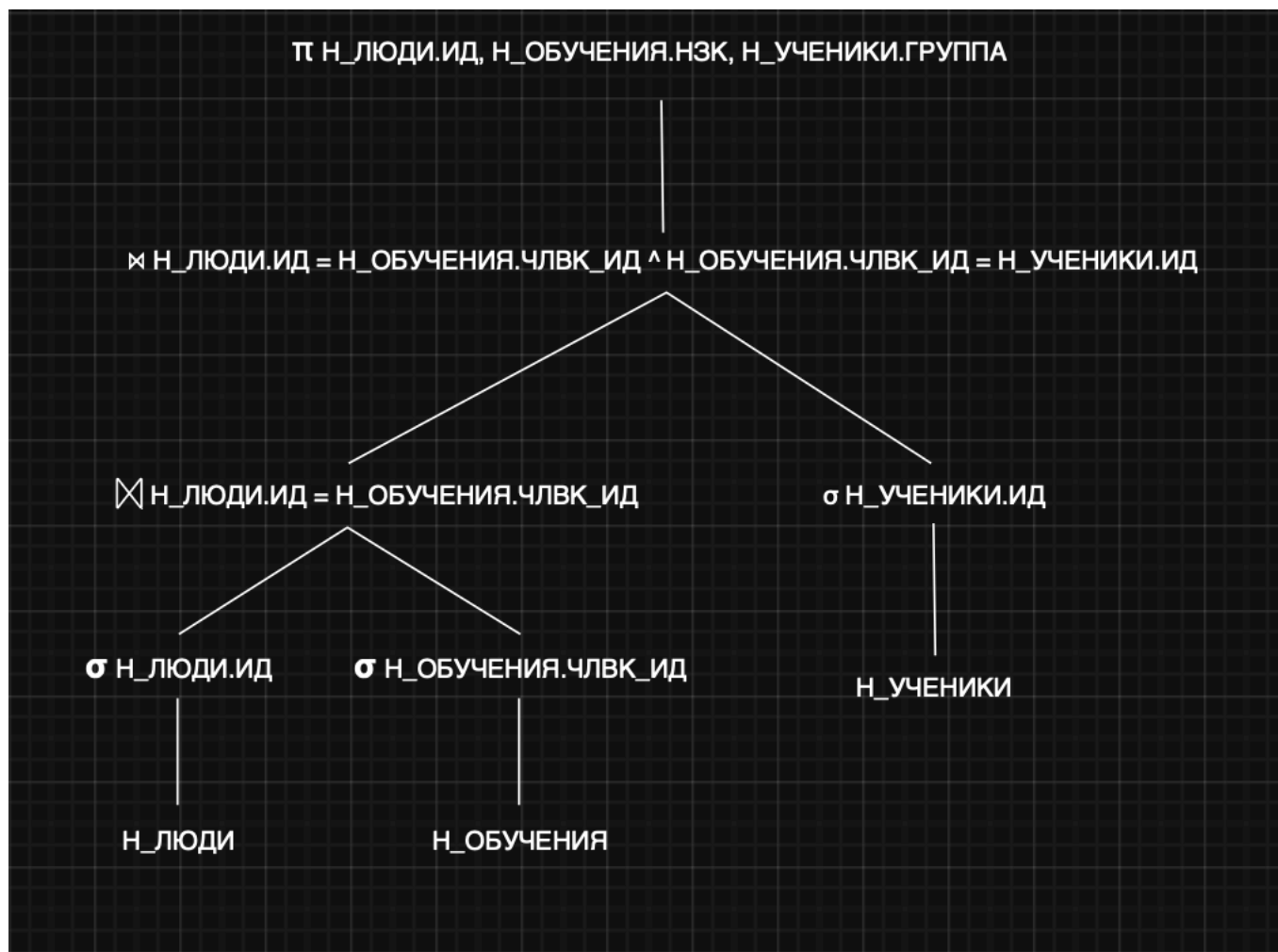
Оптимальным планом является план №3, так как все выборки проекции выполнены на раннем этапе.

2 запрос

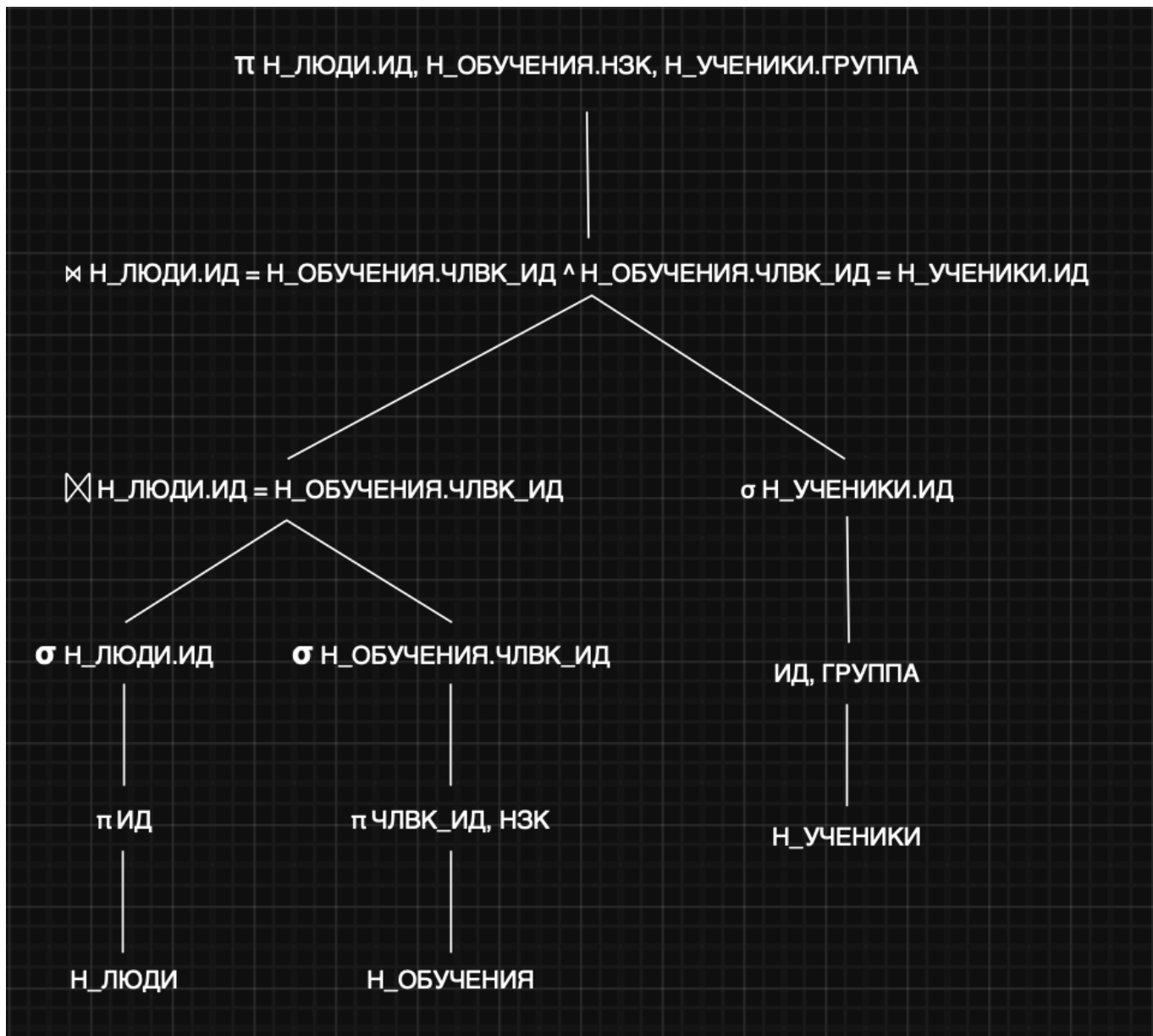
1 план



2 план



3 план



Оптимальный вариант №3 так как все выборки и проекции выполняются на ранних этапах.

Анализ

1 запрос

```

QUERY PLAN
-----
Nested Loop (cost=0.56..9.03 rows=1 width=4) (actual time=0.006..0.007 rows=0 loops=1)
-> Index Only Scan using "ЧЛВК_ПК" on "H_ЛЮДИ" (cost=0.28..4.30 rows=1 width=4) (actual time=0.005..0.005 rows=0 loops=1)
    Index Cond: ("ИД" > 163484)
    Heap Fetches: 0
-> Index Only Scan using "SYS_C003500_IFK" on "H_СЕССИЯ" (cost=0.28..4.58 rows=15 width=4) (never executed)
    Index Cond: (("ЧЛВК_ИД" = "H_ЛЮДИ"."ИД") AND ("ЧЛВК_ИД" > 100012))
    Heap Fetches: 0
Planning Time: 1.122 ms
Execution Time: 0.069 ms
(9 строк)
  
```

2 запрос

QUERY PLAN

```

Nested Loop (cost=5.35..166.82 rows=988 width=10) (actual time=1.204..1.206 rows=0 loops=1)
  -> Index Only Scan using "УЧЕН_ПК" on "Н_УЧЕНИКИ" (cost=0.29..16.39 rows=38 width=0) (actual time=0.004..0.010 rows=38 loops=1)
        Index Cond: ("ИД" < 39)
        Heap Fetches: 0
  -> Materialize (cost=5.06..138.14 rows=26 width=10) (actual time=0.031..0.031 rows=0 loops=38)
        -> Hash Join (cost=5.06..138.01 rows=26 width=10) (actual time=1.181..1.182 rows=0 loops=1)
                Hash Cond: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
                -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=5020 width=10) (actual time=0.023..0.718 rows=5020 loops=1)
                        Filter: (("ЧЛВК_ИД" IS NOT NULL) AND ("ЧЛВК_ИД" < 163484))
                        Rows Removed by Filter: 1
                -> Hash (cost=4.74..4.74 rows=26 width=4) (actual time=0.025..0.026 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=0.007..0.011 rows=40 loops=1)
                                Index Cond: ("ИД" < 100865)
                                Heap Fetches: 0

```

Planning Time: 1.507 ms

Execution Time: 1.299 ms

(17 строк)