

**Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники**

**Базы Данных
Работа с Базами Данных
Лабораторная работа №4
Вариант 8833**

Выполнил:
Студент группы Р3116
Брагин Роман Андреевич
Проверил:
Гаврилов Антон Валерьевич

г. Санкт-Петербург

2024 г

Задание:

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).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.
Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_СЕССИЯ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ИД > 100865.
б) Н_СЕССИЯ.ЧЛВК_ИД > 106059.
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.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ИД > 152862.3
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 117219.
с) Н_СЕССИЯ.ИД < 32199.
Вид соединения: LEFT JOIN.

Реализация запросов на Sql:

```
/* Запрос №1 */
SELECT "Н_ЛЮДИ"."ФАМИЛИЯ", "Н_СЕССИЯ"."ДАТА"
FROM "Н_ЛЮДИ"
LEFT JOIN "Н_СЕССИЯ"
ON "Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
WHERE "Н_ЛЮДИ"."ИД" > 100865 AND "Н_СЕССИЯ"."ЧЛВК_ИД" > 106059;
```

```
/* Запрос № 2 переделан, так как для "Н_ЛЮДИ"."ИД" > 152862 ничего не
выводит */
SELECT "Н_ЛЮДИ"."ФАМИЛИЯ" AS "ЛЮДИ_ФАМИЛИЯ",
       "Н_ВЕДОМОСТИ"."ДАТА" AS "ВЕДОМОСТИ_ДАТА",
       "Н_СЕССИЯ"."ДАТА" AS "СЕССИЯ_ДАТА"
FROM "Н_ЛЮДИ"
LEFT JOIN
"Н_ВЕДОМОСТИ" ON "Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
```

```

LEFT JOIN
  "Н_СЕССИЯ" ON "Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
WHERE ("Н_ЛЮДИ"."ИД" > 15286)
AND ("Н_ВЕДОМОСТИ"."ЧЛВК_ИД" < 117219)
AND ("Н_СЕССИЯ"."ИД" < 32199);

```

Оптимизация 1 запроса:

```
1. CREATE INDEX idx_люди_id ON "Н_ЛЮДИ" ("ИД");
```

Индекс на столбце "ИД" таблицы "Н_ЛЮДИ" ускорит отбор записей в соответствии с условием "Н_ЛЮДИ"."ИД" > 100865. Так как "ИД" используется в операции сравнения и является условием фильтрации, индекс поможет быстрее находить соответствующие записи.

```
2. CREATE INDEX idx_сессия_члвк_id ON "Н_СЕССИЯ" ("ЧЛВК_ИД");
```

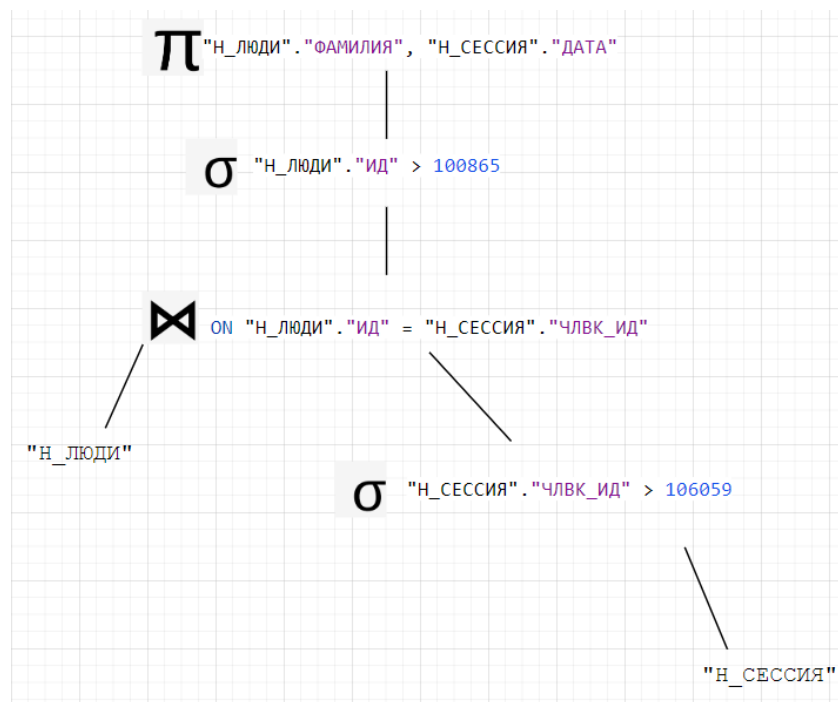
1) Ускорение отбора записей в "Н_СЕССИЯ" с условием "Н_СЕССИЯ"."ЧЛВК_ИД" > 106059, что уменьшает объем данных, просматриваемых при выполнении запроса.

2) Следствием из 1 -> Ускорение операции соединения (JOIN) с таблицей "Н_ЛЮДИ", поскольку "ЧЛВК_ИД" является ключом для соединения. Индекс поможет СУБД более эффективно находить соответствующие строки по этому столбцу.

Есть несколько планов, как можно выполнить запрос, без индексов, рассмотрим произвольных 2:

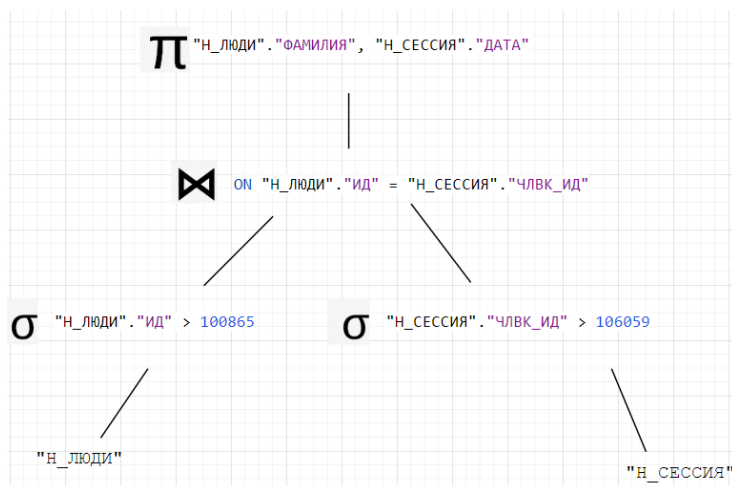
Рассмотрим 1:

- Полный скан "Н_люди".
- Полный скан по "Н_СЕССИЯ" с использованием фильтра "Н_СЕССИЯ"."ЧЛВК_ИД" > 106059
- Теперь Nested Loop Join соединяем по атрибутам "Н_люди"."ИД" и "Н_СЕССИЯ"."ЧЛВК_ИД".
- И опять делаем фильтр по "Н_люди"."ИД" > 100865.



Рассмотрим 2:

- Полный скан "Н_люди" с использованием фильтра "Н_люди". "ИД" > 100865.
- Полный скан "Н_сессия" с использованием фильтра "Н_сессия". "ЧЛВК_ИД" > 106059.
- Теперь Nested Loop Join соединяем по атрибутам "Н_люди". "ИД" и "Н_сессия". "ЧЛВК_ИД".



Оптимальный план:

План № 2, так как у нас фильтрация происходит до соединения таблиц.

План с использованием индексов idx_люди_id и idx_сессия_члвк_id:

- Вместо полного сканирования таблиц будет использоваться индексный доступ к данным.
- Nested Loop Join будет работать быстрее благодаря индексу на атрибутах "Н_сессия.ЧЛВК_ИД" и "Н_люди.ИД".
- Оптимизация производительности запроса за счет использования индексов.

EXPLAIN ANALYZE

```
Nested Loop (cost=8.46..647.37 rows=430 width=24) (actual time=0.082..0.832
rows=502 loops=1)
"  -> Bitmap Heap Scan on "Н_СЕССИЯ" (cost=8.16..85.43 rows=501 width=12)
(actual time=0.065..0.285 rows=502 loops=1)"
"    Recheck Cond: ("ЧЛВК_ИД" > 106059)"
    Heap Blocks: exact=68
"      -> Bitmap Index Scan on "SYS_C003500_IFK" (cost=0.00..8.04
rows=501 width=0) (actual time=0.046..0.046 rows=502 loops=1)"
"          Index Cond: ("ЧЛВК_ИД" > 106059)"
      -> Memoize (cost=0.29..3.21 rows=1 width=20) (actual time=0.001..0.001
rows=1 loops=502)
"          Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД""
          Cache Mode: logical
          Hits: 427 Misses: 75 Evictions: 0 Overflows: 0 Memory Usage: 9kB
"      -> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..3.20
rows=1 width=20) (actual time=0.003..0.003 rows=1 loops=75)"
"          Index Cond: (("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ИД" >
100865))"
Planning Time: 1.112 ms
Execution Time: 0.970 ms
```

Оптимизация 2 запроса:

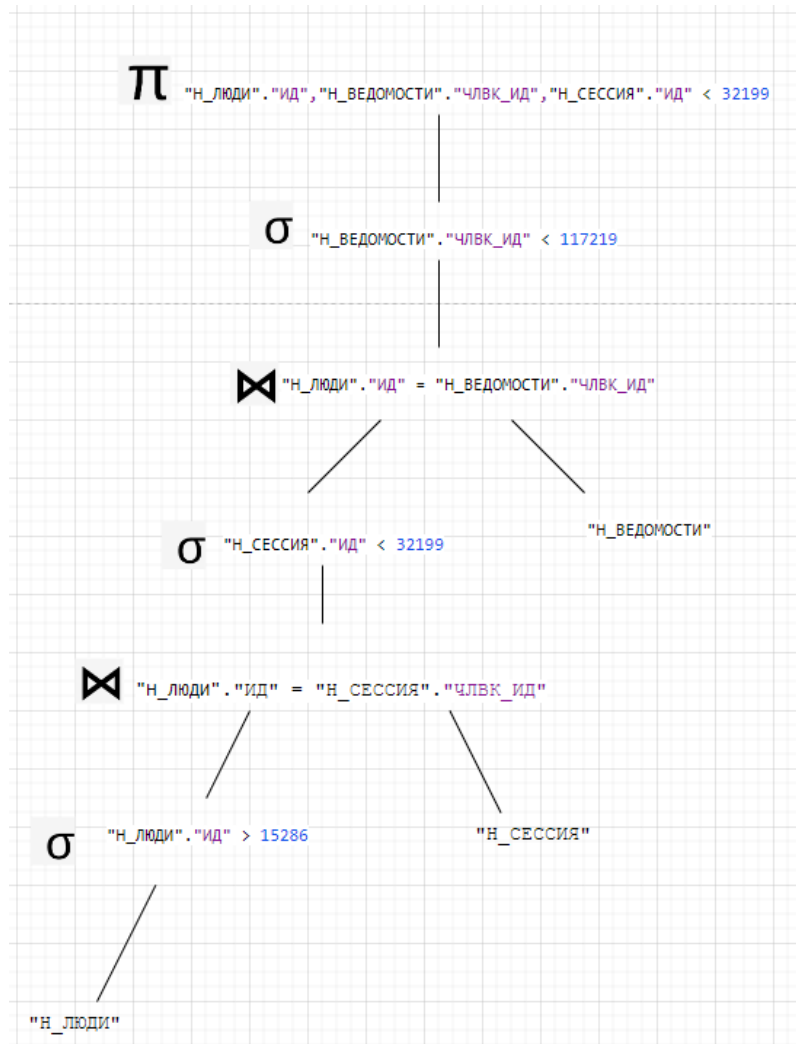
1. `CREATE INDEX idx_люди_ид ON "Н_ЛЮДИ" ("ИД");` Этот индекс поможет ускорить операции фильтрации (`WHERE "Н_ЛЮДИ"."ИД" > 15286`) и соединения (`JOIN`) на основе идентификатора в таблицах `"Н_ВЕДОМОСТИ"` и `"Н_СЕССИЯ"`.
2. `CREATE INDEX idx_ведомости_члвк_ид ON "Н_ВЕДОМОСТИ" ("ЧЛВК_ИД");` поскольку этот столбец используется для соединения с таблицей `"Н_ЛЮДИ"` и фильтрации, индекс на этом поле может значительно ускорить эти операции.
3. `CREATE INDEX idx_сессия_ид ON "Н_СЕССИЯ" ("ИД");`

Есть несколько планов, как можно выполнить запрос, без индексов, рассмотрим произвольных 2:

Рассмотрим 1:

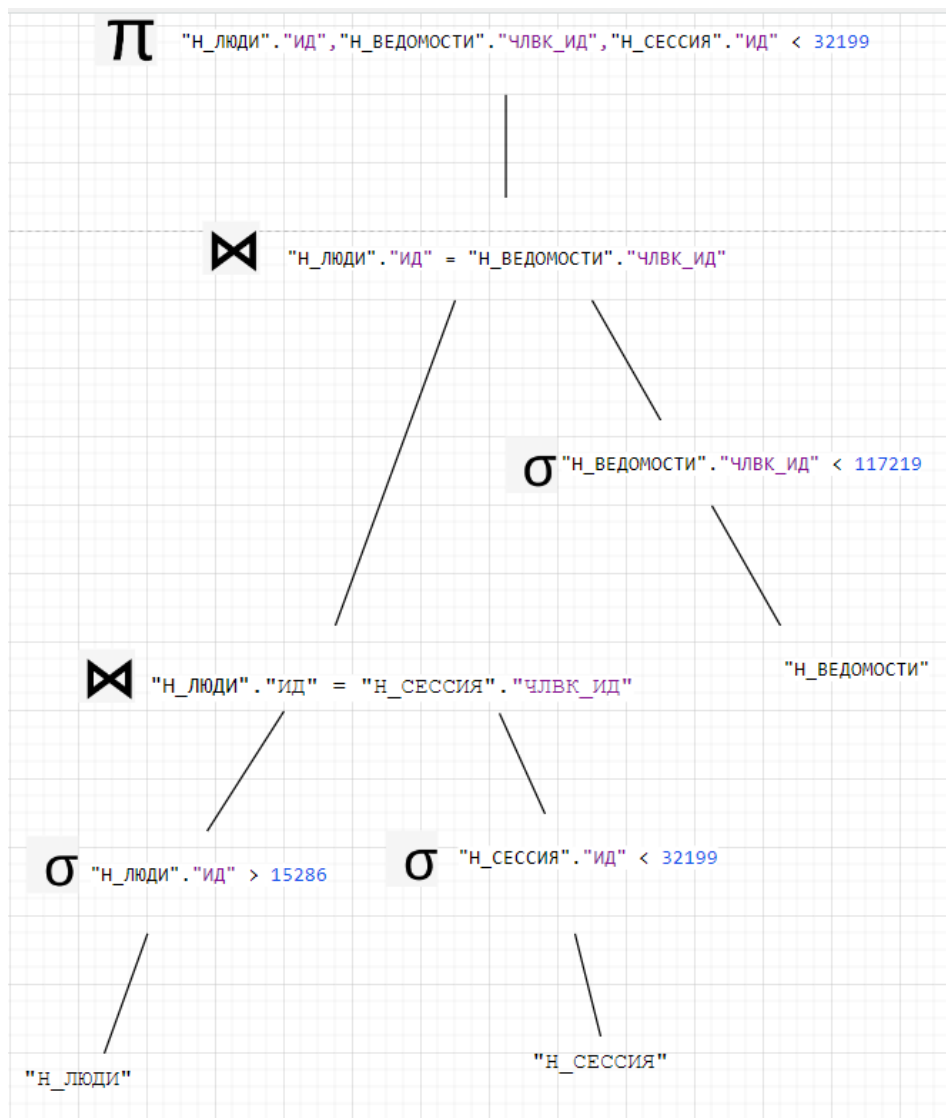
- Сначала делаем фильтр `"Н_ЛЮДИ"."ИД" > 15286` в таблице `"Н_ЛЮДИ"`.
- Теперь `Nested Loop Join` соединяем по атрибутам `"Н_ЛЮДИ"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"`.
- Теперь снова фильтр `"Н_СЕССИЯ"."ИД" < 32199`.
- Теперь `Nested Loop Join` соединяем по атрибутам `"Н_ЛЮДИ"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"`

- Теперь снова фильтрация "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" < 117219.



Рассмотрим 2:

- Сначала делаем фильтр "Н_люди"."ИД" > 15286 в таблице "Н_люди".
- Теперь снова фильтр "Н_СЕССИЯ"."ИД" < 32199.
- Теперь Nested Loop Join соединяем по атрибутам "Н_люди"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД".
- Теперь снова фильтрация "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" < 117219.
- Теперь Nested Loop Join соединяем по атрибутам "Н_люди"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"



Оптимальный план:

План № 2, так как у нас фильтрация происходит до соединения таблиц.

План с использованием индексов idx_люди_id и idx_сессия_члвк_id:

- Вместо полного сканирования таблиц будет использоваться индексный доступ к данным.
- Nested Loop Join будет работать быстрее благодаря индексу на атрибутах "Н_люди" ("ИД"), "Н_ВЕДОМОСТИ" ("ЧЛВК_ИД"), "Н_СЕССИЯ" ("ИД").
- Оптимизация производительности запроса за счет использования индексов.

EXPLAIN ANALYZE

```

Nested Loop (cost=0.60..1886.43 rows=26428 width=32) (actual time=0.375..4.967 rows=2937 loops=1)
"  Join Filter: ("Н_Люди"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД")
"    -> Nested Loop (cost=0.29..366.90 rows=3238 width=32) (actual time=0.025..2.893 rows=3238 loops=1)
"      -> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=3751 width=12) (actual time=0.011..0.666 rows=3751 loops=1)
"        Filter: ("ИД" < 32199)
"          Rows Removed by Filter: 1
"      -> Memoize (cost=0.29..0.87 rows=1 width=20) (actual time=0.000..0.000 rows=1 loops=3751)
"        Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
"        Cache Mode: logical
"        Hits: 3570 Misses: 181 Evictions: 0 Overflows: 0 Memory Usage: 22kB
"      -> Index Scan using "ЧЛВК_РК" on "Н_Люди" (cost=0.28..0.86 rows=1 width=20) (actual time=0.002..0.002
rows=1 loops=181)
"        Index Cond: (("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ИД" > 15286))
"      -> Memoize (cost=0.30..4.69 rows=13 width=12) (actual time=0.000..0.000 rows=1 loops=3238)
"        Cache Key: "Н_СЕССИЯ"."ЧЛВК_ИД"
"        Cache Mode: logical
"        Hits: 3058 Misses: 180 Evictions: 0 Overflows: 0 Memory Usage: 20kB
"      -> Index Scan using "ВЕД_ЧЛВК_ФК_ИФК" on "Н_ВЕДОМОСТИ" (cost=0.29..4.68 rows=13 width=12) (actual
time=0.001..0.002 rows=1 loops=180)
"        Index Cond: (("ЧЛВК_ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ЧЛВК_ИД" < 117219))
Planning Time: 0.611 ms
Execution Time: 5.158 ms

```

Вывод:

Выполняя данную лабораторную работу, я познакомился с индексами, оптимизацией запросов, научился строить планы выполнения запросов.