

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Лабораторная работа №4

по дисциплине «Базы данных»

Выполнил:

Лабин Макар Андреевич

Проверил:

Коновалов Арсений Антонович

г. Санкт-Петербург, 2025

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. 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:

Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 117219.

Вид соединения: LEFT JOIN.

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

Таблицы: Н_ЛЮДИ, Н_ОБУЧЕНИЯ, Н_УЧЕНИКИ.

Вывести атрибуты: Н_ЛЮДИ.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА.

Фильтры: (AND)

а) Н_ЛЮДИ.ОТЧЕСТВО < Александрович.

б) Н_ОБУЧЕНИЯ.НЗК < 001000.

с) Н_УЧЕНИКИ.ИД < 39.

Вид соединения: RIGHT JOIN.

Задание №1

Запрос на языке SQL:

```
select H_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, H_ВЕДОМОСТИ.ИД
from H_ТИПЫ_ВЕДОМОСТЕЙ
left join H_ВЕДОМОСТИ on H_ВЕДОМОСТИ.ТВ_ИД = H_ТИПЫ_ВЕДОМОСТЕЙ.ИД
where H_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'
and H_ВЕДОМОСТИ.ЧЛВК_ИД = 117219;
```

Предлагаемые индексы:

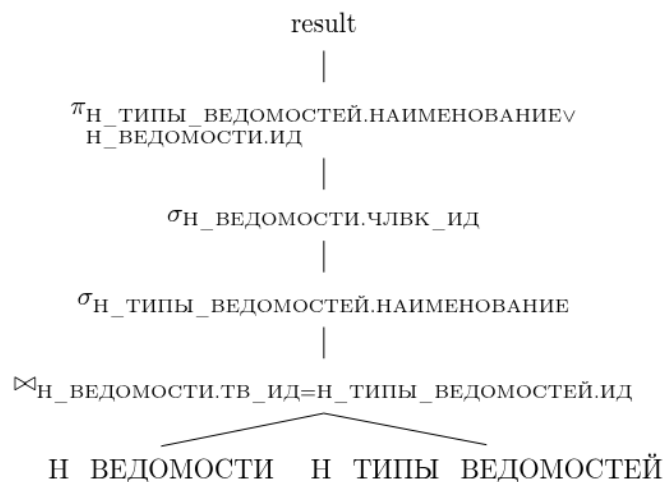
- H_ВЕДОМОСТИ.ТВ_ИД, тип hash: в запросе используется прямое сравнение в объединении двух таблиц;
- H_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, тип hash: в запросе используется прямое сравнение с константой «Экзаменационный лист»;
- H_ВЕДОМОСТИ.ЧЛВК_ИД, тип hash: в запросе используется прямое сравнение с константой 117219.

Добавление индексов в базу данных:

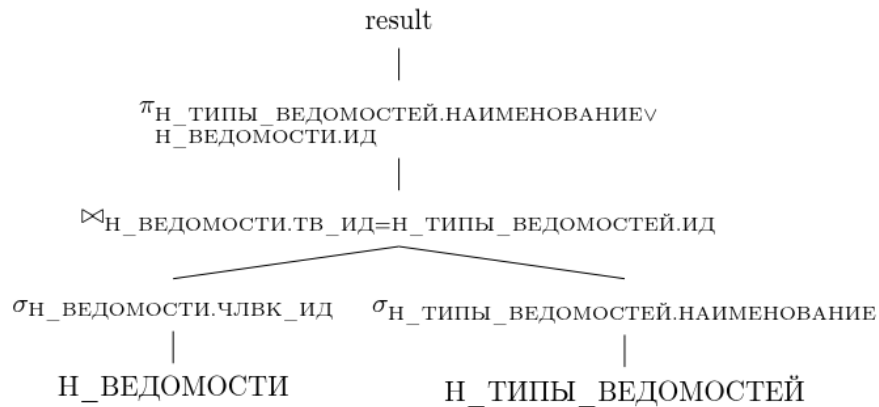
```
create index ИНДЕКС_ВЕДОМОСТИ_ТВ_ИД on H_ВЕДОМОСТИ using hash(ТВ_ИД);
create index ИНДЕКС_ТИПЫ_ВЕДОМОСТЕЙ_НАИМЕНОВАНИЕ on H_ТИПЫ_ВЕДОМОСТЕЙ
using hash(НАИМЕНОВАНИЕ);
create index ИНДЕКС_ВЕДОМОСТИ_ЧЛВК_ИД on H_ВЕДОМОСТИ using
hash(ЧЛВК_ИД);
```

Возможные планы выполнения запроса:

1. Соединяем две таблицы, затем производим выборку — **неоптимально**, поскольку в соединении участвуют все строки (*в т.ч. те, что будут отсеяны выборкой*).



2. Производим выборку в таблицах, затем соединяем их — **оптимально**, потому что в соединении участвует минимальное количество строк, необходимое для выполнения запроса.



После добавления индексов планы выполнения запросов изменятся, поскольку добавится индексное сканирование, и соединение станет быстрее благодаря индексам.

Вывод команды EXPLAIN ANALYZE:

```

QUERY PLAN
-----
Nested Loop  (cost=0.29..200.66 rows=22 width=422) (actual time=0.175..0.176 rows=0 loops=1)
  Join Filter: ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "Н_ВЕДОМОСТИ"."ТВ_ИД")
  Rows Removed by Join Filter: 31
  -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ"  (cost=0.00..1.04 rows=1 width=422) (actual time=0.023..0.024 rows=1 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text = 'Экзаменационный лист'::text)
        Rows Removed by Filter: 2
  -> Index Scan using "ВЕР_ЧЛВК_ФК_ИФК" on "Н_ВЕДОМОСТИ"  (cost=0.29..198.81 rows=65 width=8) (actual time=0.042..0.141 rows=31 loops=1)
        Index Cond: ("ЧЛВК_ИД" = 117219)
Planning Time: 1.050 ms
Execution Time: 0.233 ms
(10 строк)
  
```

Задание №2

Запрос на языке SQL:

```
select Н_ЛЮДИ.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА
from Н_ЛЮДИ
right join Н_ОБУЧЕНИЯ on Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
right join Н_УЧЕНИКИ on Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
where Н_ЛЮДИ.ОТЧЕСТВО < 'Александрович'
and Н_ОБУЧЕНИЯ.НЗК < '001000'
and Н_УЧЕНИКИ.ИД < 39;
```

Предлагаемые индексы:

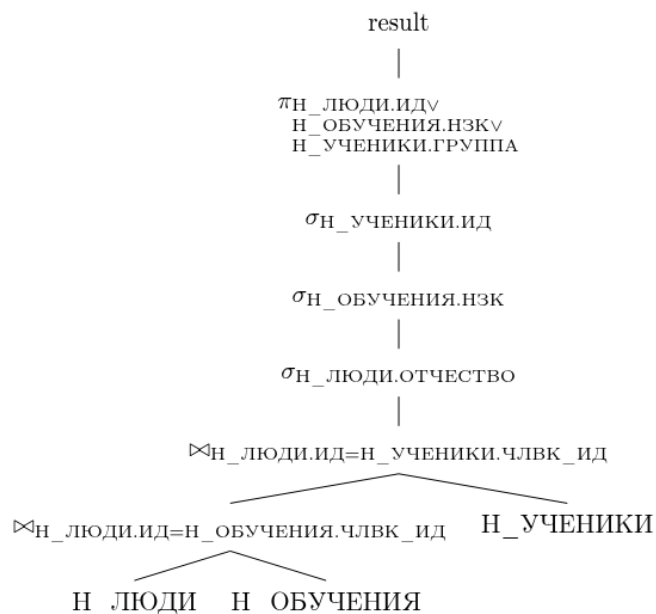
- Н_УЧЕНИКИ.ЧЛВК_ИД, тип hash: в запросе используется прямое сравнение в объединении двух таблиц;
- Н_ЛЮДИ.ОТЧЕСТВО, тип btree: в запросе используется сравнение оператором < с константой «Александрович»;
- Н_ОБУЧЕНИЯ.НЗК, тип btree: в запросе используется сравнение оператором < с константой «001000».

Добавление индексов в базу данных:

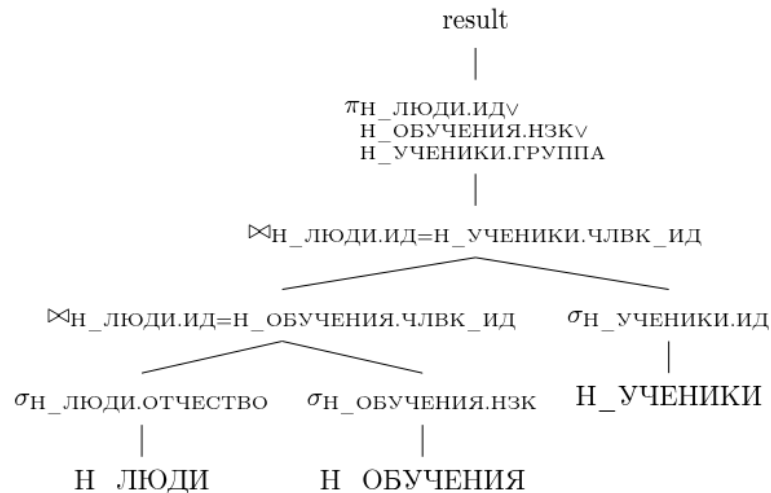
```
create index ИНДЕКС_УЧЕНИКИ_ЧЛВК_ИД on Н_УЧЕНИКИ using hash(ЧЛВК_ИД);
create index ИНДЕКС_ЛЮДИ_ОТЧЕСТВО on Н_ЛЮДИ using btree(ОТЧЕСТВО);
create index ИНДЕКС_ОБУЧЕНИЯ_НЗК on Н_ОБУЧЕНИЯ using btree(НЗК);
```

Возможные планы выполнения запроса:

1. Соединяем две таблицы, затем производим выборку — **неоптимально**, поскольку в соединении участвуют все строки (*в т.ч. те, что будут отсеяны выборкой*).



2. Производим выборку в таблицах, затем соединяем их — **оптимально**, потому что в соединении участвует минимальное количество строк, необходимое для выполнения запроса.



После добавления индексов планы выполнения запросов изменятся, поскольку добавится индексное сканирование, и соединение станет быстрее благодаря индексам.

Вывод команды EXPLAIN ANALYZE:

```

QUERY PLAN
-----
Nested Loop (cost=0.57..137.29 rows=1 width=14) (actual time=1.331..1.332 rows=0 loops=1)
-> Nested Loop (cost=0.28..128.12 rows=1 width=14) (actual time=1.330..1.331 rows=0 loops=1)
    -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=1 width=10) (actual time=1.330..1.330 rows=0 loops=1)
        Filter: (("НЗК")::text < '001000'::text)
        Rows Removed by Filter: 5021
    -> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (never executed)
        Index Cond: ("ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
        Filter: (("ОТЧЕСТВО")::text < 'Александрович'::text)
-> Index Scan using "УЧЕН_ОБУЧ_ФК_I" on "Н_УЧЕНИКИ" (cost=0.29..9.16 rows=1 width=8) (never executed)
    Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
    Filter: ("ИД" < 39)
Planning Time: 1.541 ms
Execution Time: 1.400 ms
(13 строк)
  
```

Выводы по работе

В ходе выполнения лабораторной работы я изучил теорию по индексам, оптимизации запросов и выбора плана выполнения запросов. На основе полученных знаний я составил запросы по заданию к базе данных *ucheb* сервера *helios* на языке SQL, рассмотрел различные планы выполнения запросов на конкретных примерах и аргументированно предложил выбор индексов для оптимизации составленных запросов.