

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 1541

Выполнил:

Конаныхина А.А.

P33102

Преподаватель:

Харитонова А.Е.

Санкт-Петербург, 2022 г.

Описание 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).

Реализация первого запроса:

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

Н_ТИПЫ_ВЕДОМОСТЕЙ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

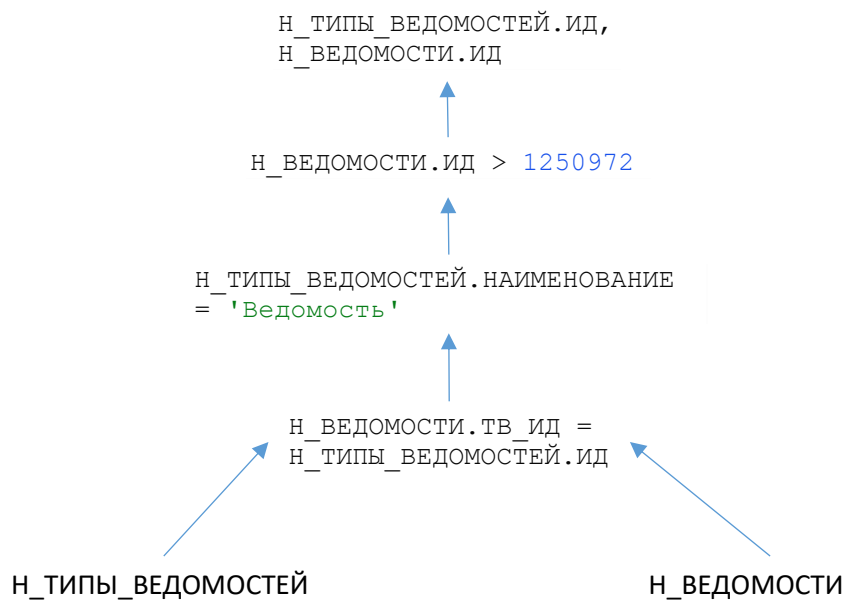
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.

б) Н_ВЕДОМОСТИ.ИД > 1250972.

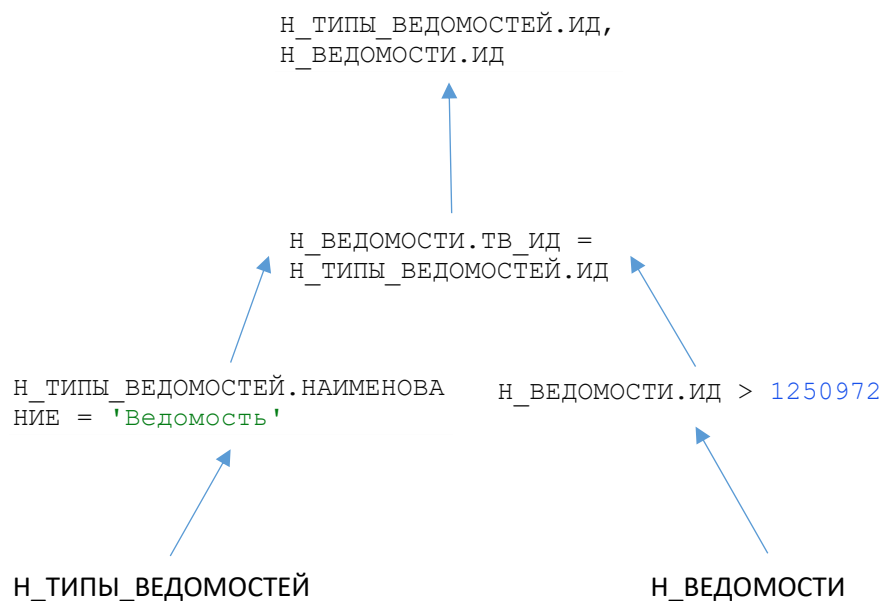
Вид соединения: INNER JOIN.

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД, Н_ВЕДОМОСТИ.ИД FROM
Н_ТИПЫ_ВЕДОМОСТЕЙ INNER JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ТВ_ИД =
Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'
AND Н_ВЕДОМОСТИ.ИД > 1250972;
```

Планы выполнения:



Оптимизируем план запроса:



Во втором плане происходит объединение только нужной выборки, а не всех сущностей. Размер промежуточных данных меньше, значит этот план является оптимальным.

Индексы:

```

CREATE INDEX ON H_ВЕДОМОСТИ USING BTREE (ИД) ;
CREATE INDEX ON H_ВЕДОМОСТИ USING HASH (ТВ_ИД) ;
CREATE INDEX ON H_ТИПЫ_ВЕДОМОСТЕЙ USING HASH (НАИМЕНОВАНИЕ) ;
CREATE INDEX ON H_ТИПЫ_ВЕДОМОСТЕЙ USING HASH (ИД) ;
  
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

Explain Analyse:

Hash Join (cost=491.20..5047.44 rows=8677 width=8) (actual time=1.683..13.097 rows=19345 loops=1)

Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")

-> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=490.15..4881.53 rows=26030 width=8) (actual time=1.624..6.794 rows=26134 loops=1)

Recheck Cond: ("ИД" > 1250972)

Heap Blocks: exact=642

-> Bitmap Index Scan on "ВЕД_ПК" (cost=0.00..483.65 rows=26030 width=0) (actual time=1.528..1.529 rows=26134 loops=1)

Index Cond: ("ИД" > 1250972)

-> Hash (cost=1.04..1.04 rows=1 width=4) (actual time=0.036..0.037 rows=1 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.023..0.024 rows=1 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text = 'Ведомость'::text)

Rows Removed by Filter: 2

Planning Time: 1.173 ms

Execution Time: 14.104 ms

Реализация второго запроса:

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

Таблицы: Н_ЛЮДИ, Н_ОБУЧЕНИЯ, Н_УЧЕНИКИ.

Вывести атрибуты: Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

а) Н_ЛЮДИ.ИМЯ > Александр.

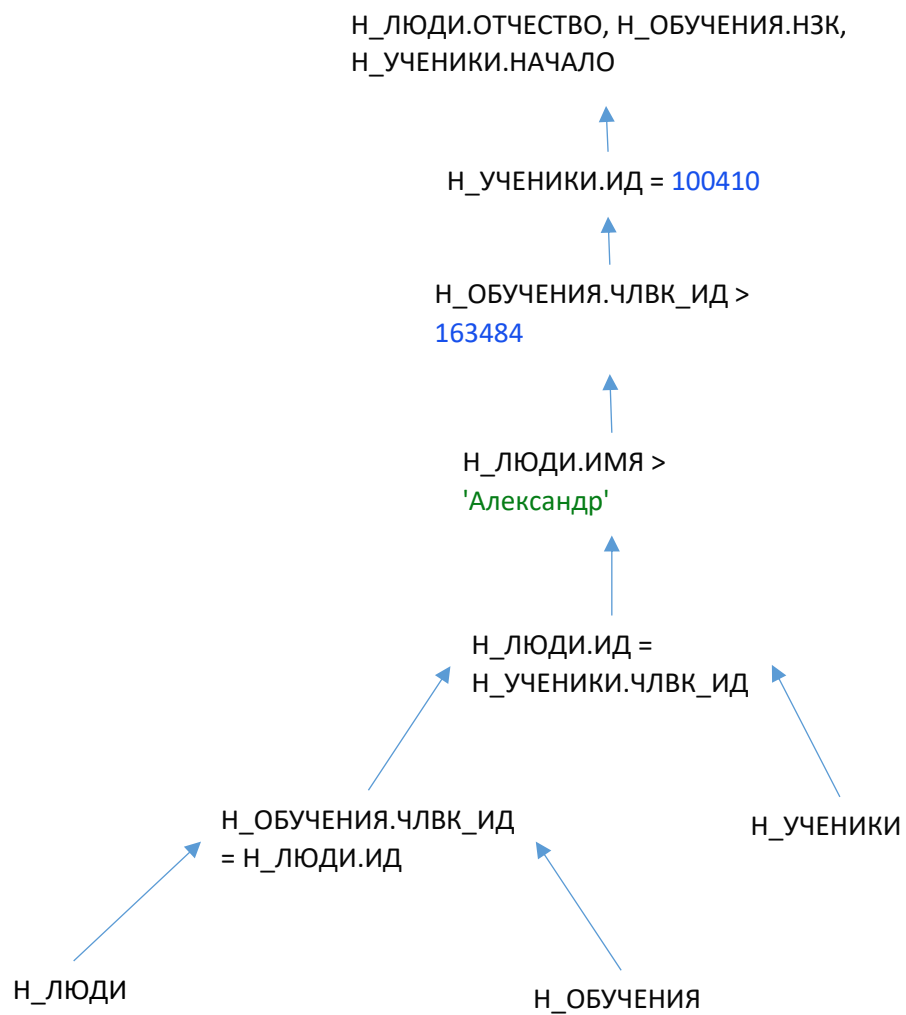
б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163484.

с) Н_УЧЕНИКИ.ИД = 100410.

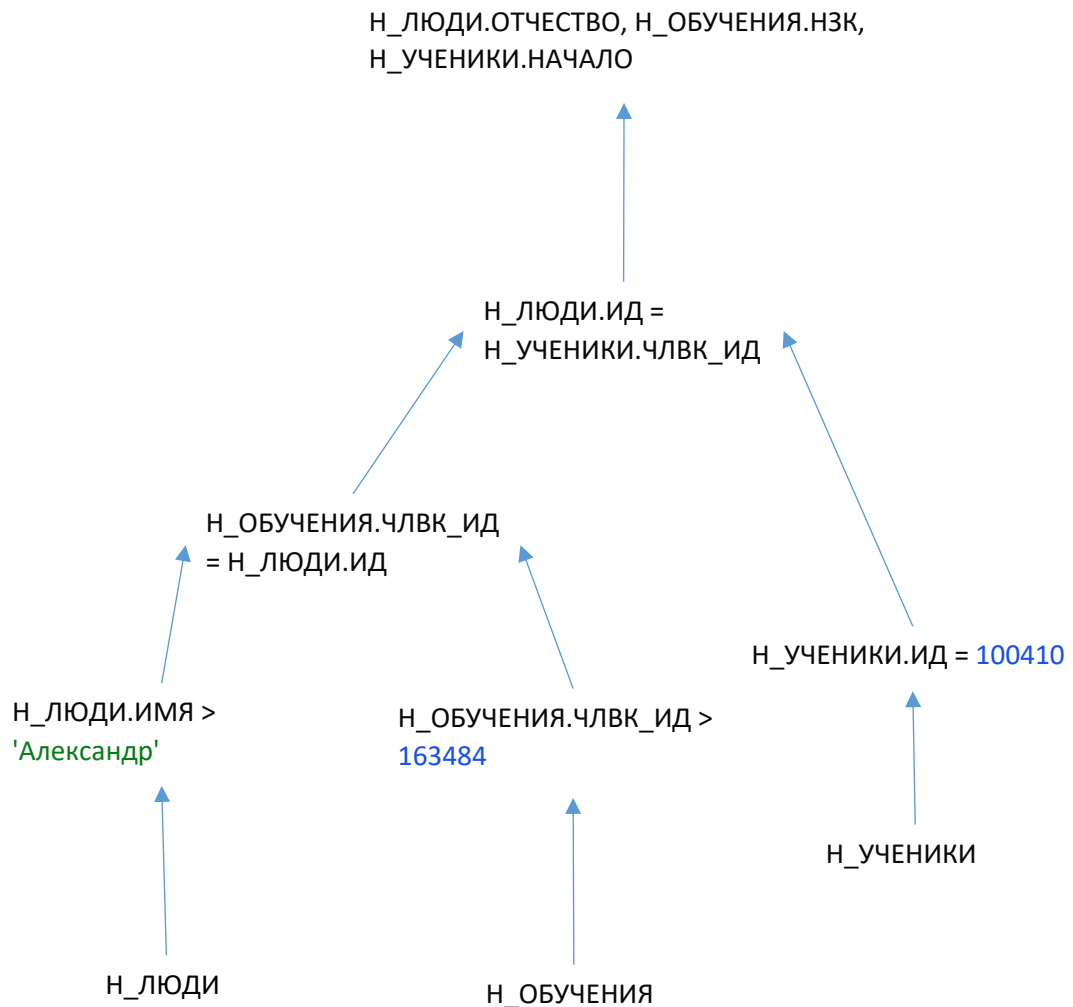
Вид соединения: LEFT JOIN.

```
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.НАЧАЛО FROM
Н_ЛЮДИ LEFT JOIN Н_ОБУЧЕНИЯ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_ЛЮДИ.ИД
LEFT JOIN Н_УЧЕНИКИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ИМЯ > 'Александр'
AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163484
AND Н_УЧЕНИКИ.ИД = 100410;
```

Планы выполнения:



Оптимизируем план запроса:



Второй план является оптимальным. Из-за того, что выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

Индексы:

```

CREATE INDEX ON Н_ЛЮДИ USING BTREE (ИМЯ) ;
CREATE INDEX ON Н_ОБУЧЕНИЯ USING BTREE (ЧЛВК_ИД) ;
CREATE INDEX ON Н_ЛЮДИ USING HASH (ИД) ;
CREATE INDEX ON Н_УЧЕНИКИ USING HASH (ИД) ;
CREATE INDEX ON Н_УЧЕНИКИ USING HASH (ЧЛВК_ИД) ;

```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

Nested Loop (cost=0.85..16.09 rows=1 width=34) (actual time=0.006..0.007 rows=0 loops=1)

Join Filter: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД")

-> Nested Loop (cost=0.56..15.18 rows=1 width=34) (actual time=0.005..0.006 rows=0 loops=1)

-> Index Scan using "ОБУЧ_ЧЛВК_FK_I" on "Н_ОБУЧЕНИЯ" (cost=0.28..6.87 rows=1 width=10) (actual time=0.004..0.005 rows=0 loops=1)

Index Cond: ("ЧЛВК_ИД" > 163484)

-> Index Scan using "ЧЛВК_PK" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (never executed)

Index Cond: ("ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")

Filter: (("ИМЯ")::text > 'Александр')::text)

-> Index Scan using "УЧЕН_ОБУЧ_FK_I" on "Н_УЧЕНИКИ" (cost=0.29..0.90 rows=1 width=12) (never executed)

Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")

Filter: ("ИД" = 100410)

Planning Time: 1.785 ms

Execution Time: 0.075 ms

Вывод:

При выполнении лабораторной работы было изучено использование индексов для ускорения обработки запросов в SQL и то, как поэтапно происходит выполнение запроса, какие планы могут использоваться и как индексы влияют на процесс.