



Вариант №1501
Лабораторная работа №4
По дисциплине
Базы Данных

Выполнил:
Студент группы Р3115
Барсуков Максим
Андреевич

Преподаватель:
Горбунов Михаил
Витальевич
Николаев Владимир
Вячеславович

1. Текст задания

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.ДАТА.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249.
Вид соединения: RIGHT 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.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.
Фильтры: (AND)
а) Н_ЛЮДИ.ИМЯ < Александр.
б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276.
с) Н_УЧЕНИКИ.ГРУППА = 3100.
Вид соединения: INNER JOIN.

2. Реализация запросов на SQL

```
-- 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.ДАТА.
-- Фильтры (AND):
-- а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.
-- б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249.
-- Вид соединения: RIGHT JOIN.
```

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ДАТА
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
RIGHT JOIN Н_ВЕДОМОСТИ ON Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = Н_ВЕДОМОСТИ.ТВ_ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость' AND
      Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249;
```

```
-- 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.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ИМЯ < Александр.
-- б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276.
-- с) Н_УЧЕНИКИ.ГРУППА = 3100.
-- Вид соединения: INNER JOIN.
```

```
SELECT Н_ЛЮДИ.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД
FROM Н_ЛЮДИ
INNER JOIN Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
INNER JOIN Н_УЧЕНИКИ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ИМЯ < 'Александр' AND
      Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276 AND
      Н_УЧЕНИКИ.ГРУППА = '3100';
```

3. Уменьшение времени выполнения 1 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

а) На таблице Н_ТИПЫ_ВЕДОМОСТЕЙ:

- Индекс на атрибуте НАИМЕНОВАНИЕ (B-tree) Это ускорит фильтрацию строк с условием "Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'".

б) На таблице Н_ВЕДОМОСТИ:

- Индекс на атрибуте ЧЛВК_ИД (B-tree) Это ускорит фильтрацию строк с условием "Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249".
- Индекс на атрибуте ТВ_ИД (B-tree) Это ускорит соединение таблиц по атрибуту ТВ_ИД.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запроса без индексов:

План 1:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ.
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249".
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД.
- Фильтрация результата соединения по условию "Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'".

План 2:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ с применением фильтра по условию "Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'".
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ.ЧЛВК_ИД > 163249".
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД.

Оптимальный план:

- План 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

При добавлении индексов планы выполнения запросов изменятся:

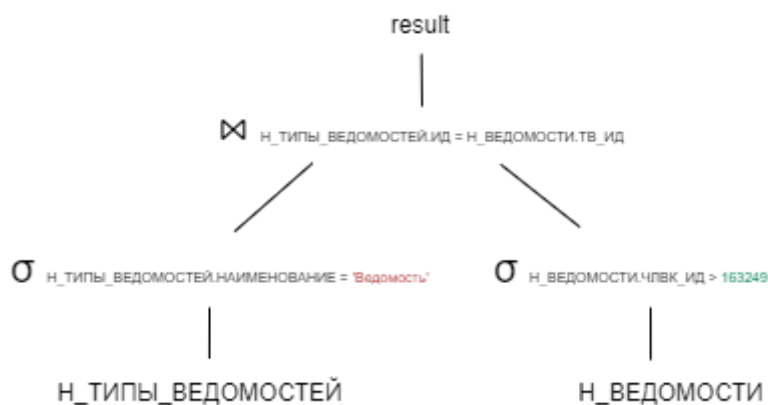
- Вместо полного скана таблиц будет использоваться индексный скан.
- Nested Loops Join станет быстрее благодаря индексу на атрибуте ТВ_ИД.

4. План выполнения 1 запроса

Будем выбирать Н_ВЕДОМОСТИ.ЧЛВК_ИД > 1, иначе нет подходящих записей.

1st QUERY PLAN

```
Nested Loop (cost=830.93..6751.65 rows=74147 width=426) (actual
time=6.784..72.728 rows=190897 loops=1)
  -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422)
      (actual time=0.011..0.016 rows=1 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text = 'Ведомость'::text)
        Rows Removed by Filter: 2
  -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=830.93..6009.14 rows=74147
width=12) (actual time=6.769..44.131 rows=190897 loops=1)
    Recheck Cond: ("ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    Filter: ("ЧЛВК_ИД" > 1)
    Heap Blocks: exact=4040
    -> Bitmap Index Scan on "ВЕД_TB_FK_I" (cost=0.00..812.40 rows=74147
width=0) (actual time=6.138..6.138 rows=190897 loops=1)
        Index Cond: ("ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
Planning Time: 0.257 ms
Execution Time: 81.498 ms
```



5. Уменьшение времени выполнения 2 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

а) На таблице Н_ЛЮДИ:

- Индекс на атрибуте ИМЯ (B-tree). Это ускорит фильтрацию строк с условием "Н_ЛЮДИ.ИМЯ < 'Александр'".

б) На таблице Н_УЧЕНИКИ:

- Индекс на атрибуте ГРУППА (B-tree). Это ускорит фильтрацию строк с условием "Н_УЧЕНИКИ.ГРУППА = '3100'".

с) На таблице Н_ОБУЧЕНИЯ:

- Индекс на атрибуте ЧЛВК_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276" и соединение таблиц по атрибуту ЧЛВК_ИД.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запросов без индексов:

План 1:

- Полный скан таблицы Н_ЛЮДИ с применением фильтра по условию "Н_ЛЮДИ.ИМЯ < 'Александр'".
- Полный скан таблицы Н_ОБУЧЕНИЯ с применением фильтра по условию "Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276".
- Соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД.
- Полный скан таблицы Н_УЧЕНИКИ с применением фильтра по условию "Н_УЧЕНИКИ.ГРУППА = '3100'".
- Соединение результатов предыдущего этапа с таблицей Н_УЧЕНИКИ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД.

План 2:

- Полный скан таблицы Н_ЛЮДИ с применением фильтра по условию "Н_ЛЮДИ.ИМЯ < 'Александр'".
- Полный скан таблицы Н_ОБУЧЕНИЯ с применением фильтра по условию "Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163276".
- Соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ с использованием Hash Join по атрибуту ЧЛВК_ИД.
- Полный скан таблицы Н_УЧЕНИКИ с применением фильтра по условию "Н_УЧЕНИКИ.ГРУППА = '3100'".
- Соединение результатов предыдущего этапа с таблицей Н_УЧЕНИКИ с использованием Hash Join по атрибуту ЧЛВК_ИД.

Оптимальный план:

- План 1, так как Nested Loops Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие, и заранее известно, что результаты фильтрации будут содержать малое количество строк.

При добавлении индексов планы выполнения запросов изменятся:

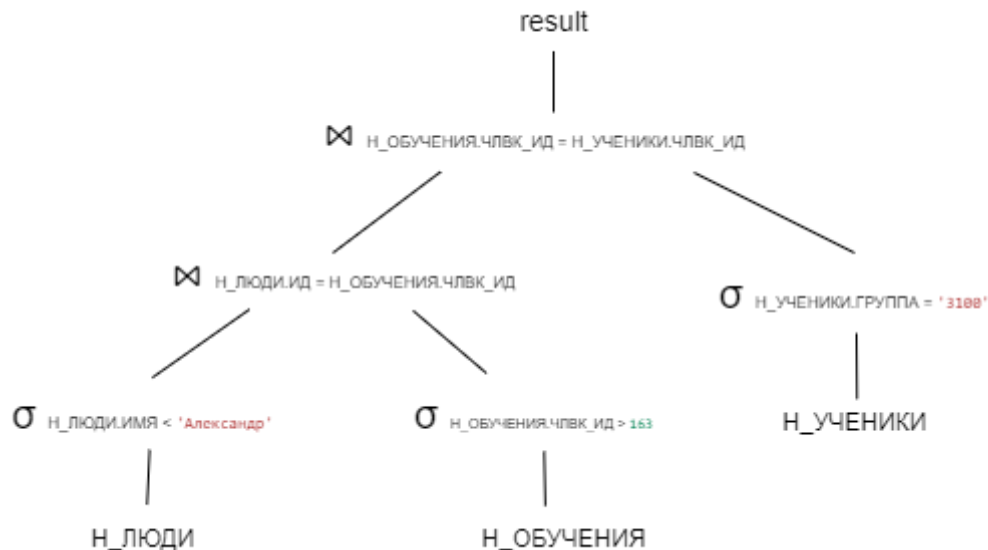
- Вместо полного скана таблиц будет использоваться индексный скан.
- Nested Loops Join станет быстрее благодаря индексам на атрибутах ЧЛВК_ИД.

6. План выполнения 2 запроса

Будем выбирать `Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 163`, иначе нет подходящих записей.

2nd QUERY PLAN

```
Nested Loop (cost=0.57..444.90 rows=8 width=21) (actual time=0.375..3.745
rows=3 loops=1)
  Join Filter: ("Н_ЛЮДИ"."ИД" = "Н_УЧЕНИКИ"."ЧЛВК_ИД")
  -> Nested Loop (cost=0.28..286.73 rows=184 width=21) (actual
time=0.025..3.010 rows=190 loops=1)
    -> Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=188 width=17)
(actual time=0.014..2.625 rows=189 loops=1)
      Filter: (("ИМЯ")::text < 'Александр'::text)
      Rows Removed by Filter: 4929
    -> Index Only Scan using "ОБУЧ_ЧЛВК_FK_I" on
"Н_ОБУЧЕНИЯ" (cost=0.28..0.64 rows=1 width=4) (actual time=0.001..0.002 rows=1
loops=189)
      Index Cond: (("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД") AND ("ЧЛВК_ИД" > 163))
      Heap Fetches: 0
  -> Index Scan using "УЧЕН_ОБУЧ_FK_I" on "Н_УЧЕНИКИ" (cost=0.29..0.85
rows=1 width=8) (actual time=0.004..0.004 rows=0 loops=190)
      Index Cond: ("ЧЛВК_ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
      Filter: (("ГРУППА")::text = '3100'::text)
      Rows Removed by Filter: 4
Planning Time: 0.711 ms
Execution Time: 3.781 ms
```



7. Вывод

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.