

Федеральное государственное автономное образовательное высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.01 – Информатика и вычислительная техника

Отчет
по лабораторной работе №4
по дисциплине «Базы данных»

Вариант 313100

Выполнил:
Колмаков Дмитрий Владимирович, Р3131
Преподаватель:
Наумова Надежда Александровна

г. Санкт-Петербург, 2023 г.

Оглавление

Задание	2
Текст задания.....	2
Ход работы.....	2
Функциональные зависимости	Error! Bookmark not defined.
Нормальные формы.....	Error! Bookmark not defined.
Денормализация	Error! Bookmark not defined.
Триггеры	Error! Bookmark not defined.
Вывод	8

Задание

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.ИД, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 3.
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 105590.
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.ИД, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД.
Фильтры (AND):
а) Н_ЛЮДИ.ИМЯ < Роман.
б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219.
с) Н_СЕССИЯ.ЧЛВК_ИД < 106059.
Вид соединения: LEFT JOIN.

Ход работы

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.ИД, Н_ВЕДОМОСТИ.ИД.
--   - Фильтры (AND):
--       а) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 3.
--       б) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 105590.
--   - Вид соединения: LEFT JOIN.
```

```
SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД", "Н_ВЕДОМОСТИ"."ИД" FROM "Н_ВЕДОМОСТИ"
LEFT JOIN "Н_ТИПЫ_ВЕДОМОСТЕЙ" ON "Н_ВЕДОМОСТИ"."ТВ_ИД" =
"Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"
WHERE "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" < 3
      AND "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = 105590;
```

```
-- 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.ИД, Н_ВЕДОМОСТИ.ДАТА, Н_СЕССИЯ.УЧГОД.
--   - Фильтры (AND):
--       а) Н_ЛЮДИ.ИМЯ < Роман.
--       б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 117219.
--       в) Н_СЕССИЯ.ЧЛВК_ИД < 106059.
--   - Вид соединения: LEFT JOIN.
```

```
SELECT "Н_ЛЮДИ"."ИД", "Н_ВЕДОМОСТИ"."ДАТА", "Н_СЕССИЯ"."УЧГОД" FROM
"Н_ВЕДОМОСТИ"
LEFT JOIN "Н_СЕССИЯ" on "Н_ВЕДОМОСТИ"."СЭС_ИД" = "Н_СЕССИЯ"."СЭС_ИД"
LEFT JOIN "Н_ЛЮДИ" on "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД"
WHERE "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 117219
      AND "Н_СЕССИЯ"."ЧЛВК_ИД" < 106059
      AND "Н_ЛЮДИ"."ИМЯ" < 'Роман';
```

Первый запрос

Уменьшение времени выполнения

Индексы, добавление которых уменьшит время выполнения запроса:

Н_ТИПЫ_ВЕДОМОСТЕЙ:

- Индекс на атрибуте ИД (B-tree): так как ИД является РК, ограничения первичного ключа автоматически создают индекс для столбцов первичного ключа (B-tree). Это ускоряет фильтрацию с условием "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" < 3.

Н_ВЕДОМОСТИ:

- Индекс на атрибуте ЧЛВК_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = 105590.
- 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 запросов без индексов:

План 1:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ.
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = 105590.
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД.
- Фильтрация результата соединения по условию "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" < 3.

План 2:

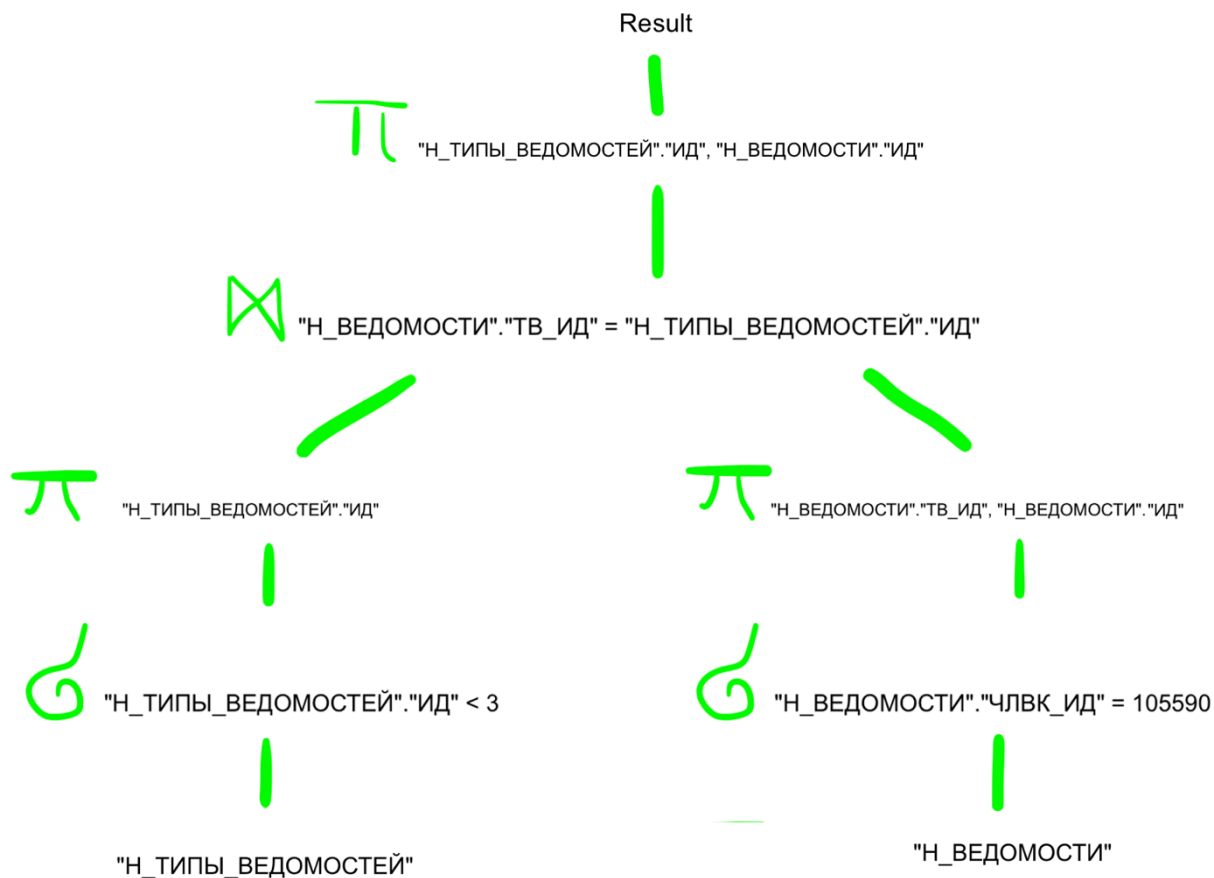
- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ с применением фильтра по условию "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" < 3.
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = 105590.
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД.

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, ускорится фильтрация.
- Nested Loop Join станет быстрее благодаря индексу на атрибуте ТВ_ИД, ускорится соединение таблиц.

План выполнения запроса



QUERY PLAN	
1	Nested Loop (cost=0.29..197.63 rows=21 width=8) (actual time=0.038..0.089 rows=27 loops=1)
2	Join Filter: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
3	Rows Removed by Join Filter: 27
4	→ Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.015..0.016 rows=2 loops=1)
5	Filter: ("ИД" < 3)
6	Rows Removed by Filter: 1
7	→ Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..195.79 rows=64 width=8) (actual time=0.010..0.029 rows=27 loops=2)
8	Index Cond: ("ЧЛВК_ИД" = 105590)
9	Planning Time: 0.236 ms
10	Execution Time: 0.129 ms

Второй запрос

Уменьшение времени выполнения

Индексы, добавление которых уменьшит время выполнения запроса:

Н_ВЕДОМОСТИ:

- Индекс на атрибуте СЭС_ИД ИД (B-tree): Это ускорит соединение таблиц по атрибуту ТВ_ИД.
- Индекс на атрибуте ЧЛВК_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 117219.

Н_СЕССИЯ:

- Индекс на атрибуте СЭС_ИД ИД (B-tree): Это ускорит соединение таблиц по атрибуту ТВ_ИД.
- Индекс на атрибуте ЧЛВК_ИД (B-tree): Это ускорит фильтрацию строк с условием "Н_СЕССИЯ"."ЧЛВК_ИД" < 106059.

Н_ЛЮДИ:

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

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_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 117219.
- Полный скан таблицы Н_ЛЮДИ с применением фильтра по условию "Н_ЛЮДИ"."ИМЯ" < 'Роман'.
- Соединение таблиц Н_ВЕДОМОСТИ и Н_ЛЮДИ с использованием Nested Loop Join по атрибуту ЧЛВК_ИД.
- Полный скан таблицы Н_СЕССИЯ с применением фильтра по условию "Н_СЕССИЯ"."ЧЛВК_ИД" < 106059.
- Соединение результатов предыдущего этапа с таблицей Н_СЕССИЯ с использованием Nested Loop Join по атрибуту СЭС_ИД.

План 2:

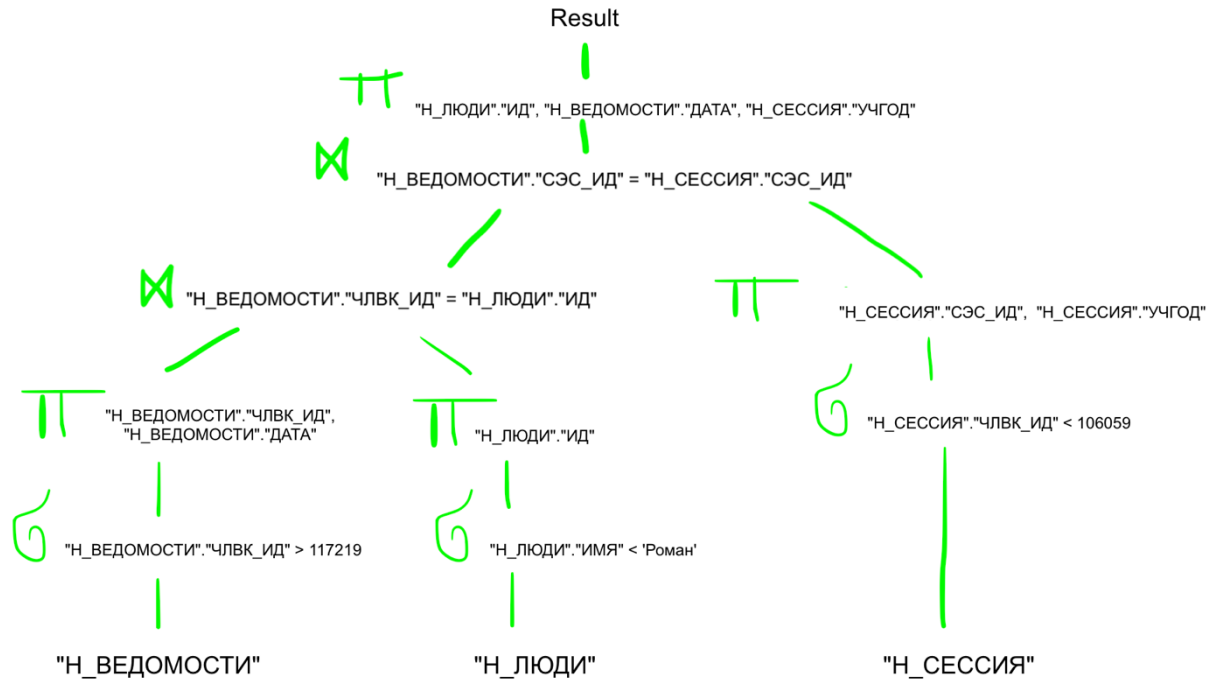
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" > 117219.
- Полный скан таблицы Н_ЛЮДИ с применением фильтра по условию "Н_ЛЮДИ"."ИМЯ" < 'Роман'.
- Соединение таблиц Н_ВЕДОМОСТИ и Н_ЛЮДИ с использованием Hash Join по атрибуту ЧЛВК_ИД.
- Полный скан таблицы Н_СЕССИЯ с применением фильтра по условию "Н_СЕССИЯ"."ЧЛВК_ИД" < 106059.
- Соединение результатов предыдущего этапа с таблицей Н_СЕССИЯ с использованием Hash Join по атрибуту СЭС_ИД.

Оптимальный план - план 2, так как Hash Join предпочтительнее Nested Loop Join в случаях, когда обрабатываемые наборы данных большие, и выполняется JOIN с использованием условия эквивалентности.

При добавлении индексов планы выполнения запросов изменятся:

- Вместо полного скана таблиц будет использоваться индексный скан, ускорится фильтрация.
- Hash Join станет быстрее благодаря индексам на атрибутах ЧЛВК_ИД и СЭС_ИД, ускорится соединение таблиц.

План выполнения запроса



	QUERY PLAN
1	Hash Join (cost=369.41..9682.13 rows=66639 width=22) (actual time=4.824..120.706 rows=111660 loops=1)
2	Hash Cond: ("Н_ВЕДОМОСТИ"."СЭС_ИД" = "Н_СЕССИЯ"."СЭС_ИД")
3	→ Hash Join (cost=217.44..7539.67 rows=151326 width=16) (actual time=3.514..88.042 rows=152421 loops=1)
4	Hash Cond: ("Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
5	→ Seq Scan on "Н_ВЕДОМОСТИ" (cost=0.00..6846.50 rows=181082 width=16) (actual time=0.004..39.866 rows=180108 loops=1)
6	Filter: ("ЧЛВК_ИД" > 117219)
7	Rows Removed by Filter: 42332
8	→ Hash (cost=163.97..163.97 rows=4277 width=4) (actual time=3.459..3.461 rows=4277 loops=1)
9	Buckets: 8192 Batches: 1 Memory Usage: 215kB
10	→ Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=4277 width=4) (actual time=0.007..2.789 rows=4277 loops=1)
11	Filter: (("ИМЯ")::text < 'Роман')::text)
12	Rows Removed by Filter: 841
13	→ Hash (cost=117.90..117.90 rows=2726 width=14) (actual time=1.283..1.284 rows=2728 loops=1)
14	Buckets: 4096 Batches: 1 Memory Usage: 160kB
15	→ Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=2726 width=14) (actual time=0.012..0.821 rows=2728 loops=1)
16	Filter: ("ЧЛВК_ИД" < 106059)
17	Rows Removed by Filter: 1024
18	Planning Time: 1.140 ms
19	Execution Time: 126.004 ms

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 скорости выполнения запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.