Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№1501**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3115

Барсуков Максим

Андреевич

*Преподаватель*:

Горбунов Михаил

Витальевич

Николаев Владимир

Вячеславович

Санкт-Петербург 2023 г.

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):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.  
   b) 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)  
   a) Н\_ЛЮДИ.ИМЯ < Александр.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.  
   c) Н\_УЧЕНИКИ.ГРУППА = 3100.  
   Вид соединения: INNER JOIN.
3. **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):      -- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.      -- b) 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)      -- a) Н\_ЛЮДИ.ИМЯ < Александр.      -- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 163276.      -- c) 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'; |

1. **Уменьшение времени выполнения 1 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:   * Индекс на атрибуте НАИМЕНОВАНИЕ (B-tree) Это ускорит фильтрацию строк с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Ведомость'".   b) 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. |

1. **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\_ТВ\_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 |

1. **Уменьшение времени выполнения 2 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ЛЮДИ:   * Индекс на атрибуте ИМЯ (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ИМЯ < 'Александр'".   b) На таблице Н\_УЧЕНИКИ:   * Индекс на атрибуте ГРУППА (B-tree). Это ускорит фильтрацию строк с условием   "Н\_УЧЕНИКИ.ГРУППА = '3100'".  c) 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. |

1. **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 |

1. **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.