Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№991**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3124

Дмитришен К. Р.

*Преподаватель*:

Кустарев Иван Павлович

Николаев Владимир Вячеславович

Санкт-Петербург 2023 г.

1. **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.  
Изменятся ли планы при добавлении индекса и как?

Для запросов 1-2 необходимо добавить в отчет вывод команды EXPLAIN ANALYZE [запрос]

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):   
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Ведомость.  
   b) Н\_ВЕДОМОСТИ.ИД < 1490007.  
   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.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ИМЯ < Роман.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 112514.  
   c) Н\_УЧЕНИКИ.ИД > 250098.  
   Вид соединения: LEFT JOIN.

1. **Реализация запросов на SQL**

|  |
| --- |
| Запрос №1:  SELECT Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД  FROM Н\_ТИПЫ\_ВЕДОМОСТЕЙ  RIGHT JOIN Н\_ВЕДОМОСТИ ON Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ВЕД\_ИД  WHERE Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > 'Ведомость'  AND Н\_ВЕДОМОСТИ.ИД < 1490007;      Запрос №2:  SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА  FROM Н\_ЛЮДИ  LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  LEFT JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  WHERE Н\_ЛЮДИ.ИМЯ < 'Роман'  AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 112514  AND Н\_УЧЕНИКИ.ИД > 250098; |

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 < 1490007". * Индекс на атрибуте ВЕД\_ИД (B-tree). Это ускорит соединение таблиц по атрибуту ВЕД\_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\_ВЕДОМОСТИ с применением фильтра по условию   «Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 1490007».   * Полное сканирование таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ. * Фильтрация строк, где НАИМЕНОВАНИЕ > «Ведомость». * Выполнение соединения с вложенным циклом между отфильтрованными результатами Н\_ТИПЫ\_ВЕДОМОСТЕЙ и Н\_ВЕДОМОСТИ..   План 2:   * Использование индекса на Н\_ВЕДОМОСТИ.ИД для фильтрации строк, где ИД < 1490007. * Использование индекса на Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ для фильтрации строк, где НАИМЕНОВАНИЕ > «Ведомость». * Использование индекса на Н\_ВЕДОМОСТИ.ВЕД\_ИД для выполнения индексированного соединения с вложенным циклом.   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. * Фильтрация и соединение будут значительно быстрее за счет использования индексов. |

1. **План выполнения 1 запроса**

|  |
| --- |
| Будем выбирать Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 1, иначе нет подходящих записей.  QUERY PLAN  ------------------------------------------------------------------------------------------------------------------  Gather (cost=1001.06..6469.10 rows=2 width=8) (actual time=28.523..31.025 rows=0 loops=1)  Workers Planned: 2  Workers Launched: 2  -> Hash Join (cost=1.06..5468.90 rows=1 width=8) (actual time=24.981..24.983 rows=0 loops=3)  Hash Cond: ("Н\_ВЕДОМОСТИ"."ВЕД\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")  -> Parallel Seq Scan on "Н\_ВЕДОМОСТИ" (cost=0.00..5224.54 rows=92683 width=8) (actual time=0.016..14.874 rows=74146 loops=3)  Filter: ("ИД" < 1490007)  Rows Removed by Filter: 0  -> Hash (cost=1.04..1.04 rows=2 width=4) (actual time=0.058..0.059 rows=2 loops=3)  Buckets: 1024 Batches: 1 Memory Usage: 9kB  -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=2 width=4) (actual time=0.045..0.046 rows=2 loops=3)  Filter: (("НАИМЕНОВАНИЕ")::text > 'Ведомость'::text)  Rows Removed by Filter: 1  Planning Time: 1.436 ms  Execution Time: 31.115 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 < 'Роман'".   b) На таблице Н\_УЧЕНИКИ:   * Индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием   "ИД > 250098".  c) На таблице Н\_ОБУЧЕНИЯ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "ЧЛВК\_ИД < 112514" 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 < 'Роман'". * Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию "ЧЛВК\_ИД < 112514". * Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "ИД > 250098". * Выполнение соединений с вложенным циклом между отфильтрованными результатами Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ и Н\_УЧЕНИКИ.   План 2:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ИМЯ < 'Роман'". * Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию "Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 112514". * Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "Н\_УЧЕНИКИ.ИД > 250098". * Использование индекса на Н\_ЛЮДИ.ЧЛВК\_ИД для выполнения индексированных соединений с вложенным циклом между отфильтрованными результатами Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ и Н\_УЧЕНИКИ.   Оптимальный план:   * План 2, так как фильтрация и соединение будут значительно быстрее за счет использования индексов.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного скана таблиц будет использоваться индексный скан. * Уменьшение количества строк, обрабатываемых при соединении, благодаря предварительной фильтрации с использованием индексов. |

1. **План выполнения 2 запроса**

|  |
| --- |
| QUERY PLAN  ------------------------------------------------------------------------------------------------------------------  Nested Loop (cost=0.85..16.71 rows=1 width=30) (actual time=0.006..0.007 rows=0 loops=1)  -> Nested Loop (cost=0.57..16.33 rows=1 width=32) (actual time=0.006..0.006 rows=0 loops=1)  -> Index Scan using "УЧЕН\_PK" on "Н\_УЧЕНИКИ" (cost=0.29..8.02 rows=1 width=8) (actual time=0.005..0.005 rows=0 loops=1)  Index Cond: ("ИД" > 250098)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (never executed)  Index Cond: ("ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД")  Filter: (("ИМЯ")::text < 'Роман'::text)  -> Index Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..0.37 rows=1 width=10) (never executed)  Index Cond: (("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД") AND ("ЧЛВК\_ИД" < 112514))  Planning Time: 1.871 ms  Execution Time: 0.056 ms |

1. **Вывод**

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов.