Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№ 6599**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3110

Чжун Цзяцзюнь

*Преподаватель*:

Гаврилов Антон Валерьевич

Санкт-Петербург 2025 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:  
   Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.  
   c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 117219.  
   Вид соединения: INNER 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.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО < Георгиевич.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590.  
   Вид соединения: RIGHT JOIN.
3. **Реализация запросов на SQL**

|  |
| --- |
| -- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  -- Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.  -- Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД.  -- Фильтры (AND):  -- a) Н\_ЛЮДИ.ОТЧЕСТВО > Георгиевич.  -- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.  -- c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 117219.  -- Вид соединения: INNER JOIN.  SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД  FROM Н\_ЛЮДИ  INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  WHERE Н\_ЛЮДИ.ОТЧЕСТВО > 'Георгиевич'  AND (Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219  OR Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 117219);  -- 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.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  -- Фильтры: (AND)  -- a) Н\_ЛЮДИ.ОТЧЕСТВО < Георгиевич.  -- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590.  -- Вид соединения: RIGHT JOIN.  SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО  FROM Н\_ЛЮДИ  RIGHT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  RIGHT JOIN Н\_УЧЕНИКИ ON Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  WHERE Н\_ЛЮДИ.ОТЧЕСТВО < 'Георгиевич'  AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590; |

1. **Уменьшение времени выполнения 1 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице **Н\_ЛЮДИ**:   * Индекс на атрибуте **ОТЧЕСТВО** (B-tree) Это ускоряет фильтрацию по условию **ОТЧЕСТВО** > 'Георгиевич', так как B-tree эффективен для операций сравнения диапазонов.   b) На таблице **Н\_ВЕДОМОСТИ**:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree) Это ускоряет фильтрацию ЧЛВК\_ИД <= 117219 и соединение по ЧЛВК\_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 <= 117219".   * Полное сканирование и сортировка таблицы Н\_ЛЮДИ по ИД с фильтром ОТЧЕСТВО > 'Георгиевич'. * Соединение через **Merge Join**.   План 2:   * Полное сканирование таблицы Н\_ВЕДОМОСТИ с фильтрацией по условию ЧЛВК\_ИД <= 117219. * Полное сканирование таблицы Н\_ЛЮДИ с фильтрацией по условию ОТЧЕСТВО > 'Георгиевич'. * Соединение таблиц через **Hash Join** по атрибуту Н\_ЛЮДИ.ИД.   План 3:   * Соединение таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ по условию ЧЛВК\_ИД * Полное сканирование и сортировка таблицы Н\_ЛЮДИ по ИД с фильтром ОТЧЕСТВО > 'Георгиевич'. * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию   "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД <= 117219".  哈希索引方法具有一些高性能与并发的特点，通常用于较大的数据列表，因此哈希索引一般在较大  Оптимальный план:   * План 2, потому что **Merge Join** требует предварительной сортировки данных, что увеличивает затраты (cost). * Без индексов оптимизатор чаще выбирает **Hash Join** для INNER JOIN с неравными условиями.   При добавлении индексов планы выполнения запросов изменятся:   * Индексы сокращают количество обрабатываемых строк. * Index Scan заменяет полное сканирование, уменьшая I/O операции. |

1. **План выполнения 1 запроса**

|  |
| --- |
| QUERY PLAN  ------------------------------------------------------------------------------------------------------------------------------------------------  Hash Join (cost=676.10..5510.22 rows=18229 width=24) (actual time=5.220..21.279 rows=18601 loops=1)  Hash Cond: ("Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")  -> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=483.96..5209.29 rows=41410 width=8) (actual time=1.694..10.192 rows=42332 loops=1)  Recheck Cond: (("ЧЛВК\_ИД" < 117219) OR ("ЧЛВК\_ИД" = 117219))  Heap Blocks: exact=1553  -> BitmapOr (cost=483.96..483.96 rows=41422 width=0) (actual time=1.512..1.513 rows=0 loops=1)  -> Bitmap Index Scan on "ВЕД\_ЧЛВК\_FK\_IFK" (cost=0.00..458.47 rows=41357 width=0) (actual time=1.506..1.507 rows=42301 loops=1)  Index Cond: ("ЧЛВК\_ИД" < 117219)  -> Bitmap Index Scan on "ВЕД\_ЧЛВК\_FK\_IFK" (cost=0.00..4.78 rows=65 width=0) (actual time=0.005..0.005 rows=31 loops=1)  Index Cond: ("ЧЛВК\_ИД" = 117219)  -> Hash (cost=163.97..163.97 rows=2253 width=24) (actual time=3.507..3.508 rows=2251 loops=1)  Buckets: 4096 Batches: 1 Memory Usage: 158kB  -> Seq Scan on "Н\_ЛЮДИ" (cost=0.00..163.97 rows=2253 width=24) (actual time=0.012..3.117 rows=2251 loops=1)  Filter: (("ОТЧЕСТВО")::text > 'Георгиевич'::text)  Rows Removed by Filter: 2867  Planning Time: 0.422 ms  Execution Time: 22.451 ms |

1. **Уменьшение времени выполнения 2 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ЛЮДИ:   * Индекс на атрибуте ОТЧЕСТВО (B-tree). Это ускорит фильтрацию строк с условием "ОТЧЕСТВО < 'Георгиевич'".   b) На таблице Н\_УЧЕНИКИ:  Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит соединения с Н\_ОБУЧЕНИЯ.  c) На таблице Н\_ОБУЧЕНИЯ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД < 105590" 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:   * Полное сканирование таблицы Н\_ОБУЧЕНИЯ с фильтрацией ЧЛВК\_ИД < 105590. * Полное сканирование таблицы Н\_ЛЮДИ с фильтрацией ОТЧЕСТВО < 'Георгиевич'. * Соединение Н\_ОБУЧЕНИЯ и Н\_ЛЮДИ через **Hash Join** по ЧЛВК\_ИД (RIGHT JOIN сохраняет все строки Н\_ОБУЧЕНИЯ). * Полное сканирование Н\_УЧЕНИКИ с фильтрацией по условию соединения. * Соединение через **Hash Join** с Н\_УЧЕНИКИ по ЧЛВК\_ИД.   План 2:   * Сортировка Н\_ОБУЧЕНИЯ по ЧЛВК\_ИД с фильтром ЧЛВК\_ИД < 105590. * Сортировка Н\_ЛЮДИ по ИД с фильтром ОТЧЕСТВО < 'Георгиевич'. * Соединение через **Merge Join** (RIGHT JOIN). * Сортировка Н\_УЧЕНИКИ по ЧЛВК\_ИД. * Соединение через **Merge Join** с Н\_УЧЕНИКИ.   Оптимальный план:   * План 1, так как в общем случае хэш-соединения работают лучше там, где можно использовать объединение. Поскольку объединение слияний должно выполнять больше сортировок, оно потребляет больше ресурсов.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного скана таблиц будет использоваться индексный скан. * Hash Join лучше обрабатывает RIGHT JOIN, сохраняя все строки из правой таблицы (Н\_ОБУЧЕНИЯ). |

1. **План выполнения 2 запроса**

|  |
| --- |
| QUERY PLAN  ------------------------------------------------------------------------------------------------------------------------------------------------  Nested Loop (cost=0.85..13.90 rows=5 width=16) (actual time=0.005..0.005 rows=0 loops=1)  -> Nested Loop (cost=0.56..12.61 rows=1 width=8) (actual time=0.004..0.005 rows=0 loops=1)  -> Index Only Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..4.30 rows=1 width=4) (actual time=0.004..0.004 rows=0 loops=1)  Index Cond: ("ЧЛВК\_ИД" < 105590)  Heap Fetches: 0  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (never executed)  Index Cond: ("ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")  Filter: (("ОТЧЕСТВО")::text < 'Георгиевич'::text)  -> Index Scan using "УЧЕН\_ОБУЧ\_FK\_I" on "Н\_УЧЕНИКИ" (cost=0.29..1.24 rows=5 width=12) (never executed)  Index Cond: ("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")  Planning Time: 2.090 ms  Execution Time: 0.101 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.