Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№52389**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3113

Бободжонов Комронджон Давронджонович

*Преподаватель*:

[Николаев Владимир Вячеславович](https://my.itmo.ru/persons/146060)

Санкт-Петербург 2024 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\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):   
   a) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < незачет.  
   b) Н\_ВЕДОМОСТИ.ИД < 1490007.  
   c) Н\_ВЕДОМОСТИ.ИД = 1490007.  
   Вид соединения: INNER 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.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ФАМИЛИЯ > Соколов.  
   b) Н\_ВЕДОМОСТИ.ИД = 1490007.  
   Вид соединения: LEFT JOIN.
3. **Реализация запросов на SQL**

|  |
| --- |
| Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям: Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ. Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД. Фильтры (AND):  a) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < незачет. b) Н\_ВЕДОМОСТИ.ИД < 1490007. c) Н\_ВЕДОМОСТИ.ИД = 1490007. Вид соединения: INNER JOIN.  SELECT Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД FROM Н\_ОЦЕНКИ  INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА WHERE Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'незачет' AND  Н\_ВЕДОМОСТИ.ИД < 1490007 AND Н\_ВЕДОМОСТИ.ИД = 1490007;  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.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА. Фильтры (AND):  a) Н\_ЛЮДИ.ФАМИЛИЯ > Соколов. b) Н\_ВЕДОМОСТИ.ИД = 1490007. Вид соединения: LEFT JOIN.  SELECT Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА FROM Н\_ВЕДОМОСТИ  LEFT JOIN Н\_ЛЮДИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  LEFT JOIN Н\_СЕССИЯ ON Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД WHERE Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов'  AND Н\_ВЕДОМОСТИ.ИД = 1490007; |

1. **Уменьшение времени выполнения 1 запроса**

|  |
| --- |
| Индексы , добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ОЦЕНКИ:   * Индекс на атрибуте ПРИМЕЧАНИЕ B-tree Это ускорит фильтрацию строк с условием Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'незачет'.   b) На таблице Н\_ВЕДОМОСТИ:   * Индекс на атрибуте ИД B-tree Это ускорит фильтрацию строк с условием " Н\_ВЕДОМОСТИ.ИД > 163249". * Индекс на атрибуте ОЦЕНКА Hash Это ускорит соединение таблиц по атрибуту ОЦЕНКА..   Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.  Index:  CREATE INDEX idx\_Marks  ON Н\_ОЦЕНКИ USING btree (ПРИМЕЧАНИЕ);  CREATE INDEX idx\_Vedomosti\_id  ON Н\_ВЕДОМОСТИ. USING btree (ИД);  CREATE INDEX idx\_Osenka\_id  ON ОЦЕНКА USING hash (ИД);  Возможные планы выполнения запроса без индексов:  План 1:   * Полный скан таблицы Н\_ОЦЕНКИ. * Полный скан таблицы Н\_ОЦЕНКИ с применением фильтра по условию   " Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'незачет' ".   * Соединение таблиц с использованием Nested Loops Join по атрибуту Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА. * Фильтрация результата соединения по условию " Н\_ВЕДОМОСТИ.ИД < 1490007 AND Н\_ВЕДОМОСТИ.ИД = 1490007;'".   План 2:   * Взять проекцию Н\_ОЦЕНКИ атрибуты Н\_ОЦЕНКИ.КОД, Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ * Полный скан таблицы Н\_ОЦЕНКИ.КОД с применением фильтра по условию " Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'незачет' ". * Взять проекцию Н\_ВЕДОМОСТИ атрибуты Н\_ВЕДОМОСТИ.ИД,Н\_ВЕДОМОСТИ.ОЦЕНКА Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию " Н\_ВЕДОМОСТИ.ИД < 1490007 AND Н\_ВЕДОМОСТИ.ИД = 1490007;". * Соединение таблиц с использованием Nested Loops Join по атрибуту Н\_ОЦЕНКИ.КОД = Н\_ВЕДОМОСТИ.ОЦЕНКА.   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**

|  |
| --- |
| 1st QUERY PLAN  ------------------------------------------------------------------------------------------------------------------  Nested Loop (cost=0.42..9.56 rows=1 width=9) (actual time=0.021..0.022 rows=0 loops=1)  Join Filter: (("Н\_ОЦЕНКИ"."КОД")::text = ("Н\_ВЕДОМОСТИ"."ОЦЕНКА")::text)  -> Seq Scan on "Н\_ОЦЕНКИ" (cost=0.00..1.11 rows=1 width=5) (actual time=0.013..0.017 rows=2 loops=1)  Filter: (("ПРИМЕЧАНИЕ")::text < 'незачет'::text)  Rows Removed by Filter: 7  -> Index Scan using "ВЕД\_PK" on "Н\_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=10) (actual time=0.001..0.001 rows=0 loops=2)  Index Cond: (("ИД" < 1490007) AND ("ИД" = 1490007))  Planning Time: 0.658 ms  Execution Time: 0.048 ms  Plan1:    Plan 2: |

1. **Уменьшение времени выполнения 2 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ЛЮДИ:   * Индекс на атрибуте ФАМИЛЯ (B-tree). Это ускорит фильтрацию строк с условием " Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов'".   b) На таблице Н\_ВЕДОМОСТИ:   * Индекс на атрибуте ИД (Hash). Это ускорит фильтрацию строк с условием   “Н\_ВЕДОМОСТИ.ИД = 1490007".  c) На таблице Н\_СЕССИЯ:   * Индекс на атрибуте ЧЛВК\_ИД (Hash). Это ускорит соединение таблиц по атрибуту ЧЛВК\_ИД.   Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.  Index:  CREATE INDEX idx\_Marks  ON Н\_ЛЮДИ USING btree (ФАМИЛЯ);  CREATE INDEX idx\_Vedomosti\_id  ON Н\_ВЕДОМОСТИ. USING hash (ИД);    CREATE INDEX idx\_Sessiya\_id  ON Н\_СЕССИЯ USING hash (ИД);  Возможные планы выполнения запросов без индексов:  План 1:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию " Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов''". * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию " Н\_ВЕДОМОСТИ.ИД = 1490007". * Соединение таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ с использованием Nested Loops Join по атрибуту Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД. * Полный скан таблицы Н\_СЕССИЯ * Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Nested Loops Join по атрибуту Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.   План 2:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию " Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов''". * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию " Н\_ВЕДОМОСТИ.ИД = 1490007". * Соединение таблиц Н\_ВЕДОМОСТИ и Н\_ЛЮДИ с использованием Hash Join по атрибуту Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД. * Полный скан таблицы Н\_СЕССИЯ * Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Nested Loops Join по атрибуту Н\_СЕССИЯ.ЧЛВК\_ИД = Н\_ВЕДОМОСТИ.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**

|  |
| --- |
| 2nd QUERY PLAN  ------------------------------------------------------------------------------------------------------------------  Nested Loop Left Join (cost=5.12..62.79 rows=1 width=16) (actual time=0.044..0.045 rows=0 loops=1)  -> Nested Loop (cost=0.70..16.75 rows=1 width=8) (actual time=0.043..0.044 rows=0 loops=1)  -> Index Scan using "ВЕД\_PK" on "Н\_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=4) (actual time=0.009..0.010 rows=1 loops=1)  Index Cond: ("ИД" = 1490007)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.028..0.028 rows=0 loops=1)  Index Cond: ("ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  Filter: (("ФАМИЛИЯ")::text > 'Соколов'::text)  Rows Removed by Filter: 1  -> Bitmap Heap Scan on "Н\_СЕССИЯ" (cost=4.42..45.86 rows=18 width=12) (never executed)  Recheck Cond: ("ЧЛВК\_ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  -> Bitmap Index Scan on "SYS\_C003500\_IFK" (cost=0.00..4.42 rows=18 width=0) (never executed)  Index Cond: ("ЧЛВК\_ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  Planning Time: 0.997 ms  Execution Time: 0.120 ms  Plan1:  Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА.    Plan2: |
|  |
|  |

1. **Вывод**

В ходе выполнения лабораторной работы я научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации запросов.