Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№97**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы Р3108

Коломиец Никита Сергеевич

*Преподаватель*:

Егошин Алексей

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.ИД, Н\_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

* 1. Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 1.
  2. Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08.
  3. Н\_ВЕДОМОСТИ.ДАТА > 1998-01-05.

Вид соединения: LEFT JOIN.

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.

Фильтры: (AND)

* 1. Н\_ЛЮДИ.ИМЯ = Ярослав.
  2. Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.

Вид соединения: RIGHT JOIN.

1. **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.ДАТА.  -- Фильтры (AND):  -- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 1.  -- b) Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08.  -- c) Н\_ВЕДОМОСТИ.ДАТА > 1998-01-05.  SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД",  "Н\_ВЕДОМОСТИ"."ДАТА"  FROM "Н\_ВЕДОМОСТИ"  LEFT JOIN "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" ON  "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"  WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" > 1  AND "Н\_ВЕДОМОСТИ"."ДАТА" > '2022-06-08'  AND "Н\_ВЕДОМОСТИ"."ДАТА" > '1998-01-05';     -- 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.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  -- Фильтры: (AND)  -- a) Н\_ЛЮДИ.ИМЯ = Ярослав.  -- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.  -- Вид соединения: RIGHT JOIN.     SELECT "Н\_ЛЮДИ"."ФАМИЛИЯ",  "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД",  "Н\_УЧЕНИКИ"."ИД"  FROM "Н\_ОБУЧЕНИЯ"  RIGHT JOIN "Н\_ЛЮДИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  RIGHT JOIN "Н\_УЧЕНИКИ" ON "Н\_УЧЕНИКИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  WHERE "Н\_ЛЮДИ"."ИМЯ" = 'Ярослав'  AND "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = 105590; |

1. **Уменьшение времени выполнения 1 запроса**

|  |
| --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:   * Индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" > 1   b) На таблице Н\_ВЕДОМОСТИ:   * Индекс на атрибуте ДАТА (B-tree). Это ускорит фильтрацию строк с условиями "Н\_ВЕДОМОСТИ"."ДАТА" > '2022-06-08' И   "Н\_ВЕДОМОСТИ"."ДАТА" > '1998-01-05'   * 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 с применением фильтра по условию   Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 1   * Полный скан таблицы Н\_ВЕДОМОСТИ с приминением фильтров по условию "Н\_ВЕДОМОСТИ"."ДАТА" > '2022-06-08' и "Н\_ВЕДОМОСТИ"."ДАТА" > '1998-01-05' * Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД   План 2:   * Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ * Полный скан таблицы Н\_ВЕДОМОСТИ с приминением фильтров по условию "Н\_ВЕДОМОСТИ"."ДАТА" > '2022-06-08' и "Н\_ВЕДОМОСТИ"."ДАТА" > '1998-01-05' * Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД * Фильтрация результата по условию Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД > 1   Оптимальный план:   * План 1, 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**

|  |
| --- |
| QUERY PLAN  -------------------------------------------------------------------------------  Nested Loop (cost=0.29..8.25 rows=1 width=12) (actual time=0.007..0.008 rows=0 loops=1)  Join Filter: ("Н\_ВЕДОМОСТИ"."ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")  -> Index Scan using "ВЕД\_ДАТА\_I" on "Н\_ВЕДОМОСТИ" (cost=0.29..7.20 rows=1 width=12) (actual time=0.006..0.006 rows=0 loops=1)  Index Cond: (("ДАТА" > '2022-06-08 00:00:00'::timestamp without time zone) AND ("ДАТА" > '1998-01-05 00:00:00'::timestamp without time zone))  -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (never executed)  Filter: ("ИД" > 1)  Planning Time: 1.229 ms  Execution Time: 0.081 ms  (8 строк) |

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" = 'Ярослав'   b) На таблице Н\_УЧЕНИКИ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит соединение таблиц по атрибуту ЧЛВК\_ИД   c) На таблице Н\_ОБУЧЕНИЯ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит соединение таблиц по атрибуту ЧЛВК\_ИД и филтрацию строк с условием "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = 105590   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" = 'Ярослав' * Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = 105590 * Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД * Полный скан таблицы Н\_УЧЕНИКИ * Соединение результатов предыдущего соединения 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" = 'Ярослав' * Полный скан таблицы Н\_ОБУЧЕНИЯ с применением фильтра по условию "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = 105590 * Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЯ с использованием Hash Join по атрибуту ЧЛВК\_ИД * Полный скан таблицы Н\_УЧЕНИКИ * Соединение результатов предыдущего соединения 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**

|  |
| --- |
| QUERY PLAN  --------------------------------------------------------------------------------  Nested Loop (cost=4.89..35.61 rows=5 width=24) (actual time=0.070..0.071 rows=0 loops=1)  -> Nested Loop (cost=0.56..12.61 rows=1 width=24) (actual time=0.069..0.070 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=0.018..0.018 rows=1 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=20) (actual time=0.049..0.050 rows=0 loops=1)  Index Cond: ("ИД" = 105590)  Filter: (("ИМЯ")::text = 'Ярослав'::text)  Rows Removed by Filter: 1  -> Bitmap Heap Scan on "Н\_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=8) (never executed)  Recheck Cond: ("ЧЛВК\_ИД" = 105590)  -> Bitmap Index Scan on "УЧЕН\_ОБУЧ\_FK\_I" (cost=0.00..4.32 rows=5 width=0) (never executed)  Index Cond: ("ЧЛВК\_ИД" = 105590)  Planning Time: 0.969 ms  Execution Time: 0.128 ms  (15 строк) |

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.