Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№7**

Лабораторная работа **№4**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3112

Марьин Григорий Алексеевич

*Преподаватель*:

Кустарев Иван Павлович

Санкт-Петербург 2025г.

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.ЧЛВК\_ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Ведомость.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
   c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 153285.  
   Вид соединения: LEFT 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.ОТЧЕСТВО < Сергеевич.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.  
   Вид соединения: RIGHT JOIN.
3. **Реализация первого запроса на SQL**

SELECT  
 Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД,  
 Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  
FROM  
 Н\_ТИПЫ\_ВЕДОМОСТЕЙ  
LEFT JOIN  
 Н\_ВЕДОМОСТИ  
ON  
 Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ТВ\_ИД  
WHERE  
 Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < 'Ведомость'  
 AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249  
 AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 153285;

1. **Уменьшение времени выполнения первого 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**

|  |
| --- |
| QUERY PLAN  -------------------------------------------------------------------------------  Nested Loop (cost=0.29..8.32 rows=1 width=8) (actual time=0.026..0.027 rows=0 loops=1)  Join Filter: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД")  -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.025..0.026 rows=0 loops=1)  Filter: (("НАИМЕНОВАНИЕ")::text < 'Ведомость'::text)  Rows Removed by Filter: 3  -> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..7.27 rows=1 width=8) (never executed)  Index Cond: (("ЧЛВК\_ИД" > 163249) AND ("ЧЛВК\_ИД" > 153285))  Planning Time: 0.363 ms  Execution Time: 0.069 ms |

1. **Реализация второго запроса на SQL**

SELECT  
 Н\_ЛЮДИ.ИМЯ,  
 Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД,  
 Н\_УЧЕНИКИ.ИД  
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.ИД  
WHERE  
 Н\_ЛЮДИ.ОТЧЕСТВО < 'Сергеевич'  
 AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276;

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.ОТЧЕСТВО < 'Сергеевич".   b) На таблице Н\_ОБУЧЕНИЯ:   * Индекс на атрибуте ЧЛВК\_ИД (Hash Index). 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.ОТЧЕСТВО < 'Сергеевич ". * 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.   План 2:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ИМЯ < 'Сергеевич". * 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.   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=0.85..20.93 rows=1 width=21) (actual time=0.095..0.096 rows=0 loops=1)  -> Nested Loop (cost=0.56..12.61 rows=1 width=17) (actual time=0.046..0.048 rows=1 loops=1)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=17) (actual time=0.020..0.021 rows=1 loops=1)  Index Cond: ("ИД" = 163276)  Filter: (("ОТЧЕСТВО")::text < 'Сергеевич'::text)  -> Index Only Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..4.30 rows=1 width=4) (actual time=0.022..0.022 rows=1 loops=1)  Index Cond: ("ЧЛВК\_ИД" = 163276)  Heap Fetches: 0  -> Index Only Scan using "УЧЕН\_PK" on "Н\_УЧЕНИКИ" (cost=0.29..8.30 rows=1 width=4) (actual time=0.047..0.047 rows=0 loops=1)  Index Cond: ("ИД" = 163276)  Heap Fetches: 0  Planning Time: 1.156 ms  Execution Time: 0.134 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.