НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной инженерии и компьютерной техники

Базы данных

Лабораторная работа № 4

Выполнил студент:

Лабор Тимофей Владимирович

Группа № P3125

Преподаватель:

Бострикова Дарья Константиновна

г. Санкт-Петербург

2024

Оглавление

[**Вариант:** 3](#_Toc155691970)

[**Задание:** 3](#_Toc155691971)

[**Отчет:** 3](#_Toc155691972)

[**Вывод:** 9](#_Toc155691973)

[**Список литературы:** **Ошибка! Закладка не определена.**](#_Toc155691974)

## **Вариант:**

2508

## **Задание:**

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\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.ИД.  
   Фильтры (AND):   
   a) Н\_ЛЮДИ.ИД = 152862.  
   b) Н\_СЕССИЯ.ИД > 27640.  
   c) Н\_СЕССИЯ.ИД < 1975.  
   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.ИД, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.ГРУППА.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО > Владимирович.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.  
   Вид соединения: LEFT JOIN.

## **Отчет:**

1. **Реализация запросов на SQL**

|  |
| --- |
| -- Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.  -- Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.ИД.  -- Фильтры (AND):  -- a) Н\_ЛЮДИ.ИД = 152862.  -- b) Н\_СЕССИЯ.ИД > 27640.  -- c) Н\_СЕССИЯ.ИД < 1975.  -- Вид соединения: LEFT JOIN.  SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.ИД  FROM Н\_ЛЮДИ  LEFT JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД  WHERE Н\_ЛЮДИ.ИД = 152862      AND Н\_СЕССИЯ.ИД > 27640      AND Н\_СЕССИЯ.ИД < 1975;  -- 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.НЗК, Н\_УЧЕНИКИ.ГРУППА.  -- Фильтры: (AND)  -- a) Н\_ЛЮДИ.ОТЧЕСТВО > Владимирович.  -- b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.  -- Вид соединения: LEFT JOIN.  SELECT Н\_ЛЮДИ.ИД, 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.ОТЧЕСТВО > 'Владимирович'      AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276; |

1. **Уменьшение времени выполнения 1 запроса**

|  |  |
| --- | --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  a) На таблице Н\_ЛЮДИ:   * Было бы полезно иметь индекс на атрибуте ИД (B-tree), это ускоряет фильтрацию строк с условием "Н\_ЛЮДИ.ИД = 152862", и такой индекс уже есть в базе данных.   b) На таблице Н\_СЕССИЯ:   * Индекс на атрибуте ИД (B-tree), это ускорит фильтрацию строк с условием "Н\_СЕССИЯ.ИД > 27640 AND Н\_СЕССИЯ.ИД < 1975". * Было бы полезно иметь индекс на атрибуте ЧЛВК\_ИД (B-tree), это может ускорить соединение с таблицей Н\_ЛЮДИ по ЧЛВК\_ИД, и такой индекс уже есть в базе данных.   Возможные планы выполнения запроса (без индексов): | |
| План 1:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию “Н\_ЛЮДИ.ИД=152862”. * Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию “Н\_СЕССИЯ.ИД > 27640 AND Н\_СЕССИЯ.ИД < 1975”. * Соединение таблиц с использованием Nested Loops Join по условию Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД.   Изображение выглядит как текст, диаграмма, линия, снимок экрана  Автоматически созданное описание | План 2:   * Соединение таблиц с использованием Nested Loops Join по условию Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД. * Фильтрация результата соединения по условию “Н\_ЛЮДИ.ИД=152862 AND Н\_СЕССИЯ.ИД > 27640 AND Н\_СЕССИЯ.ИД < 1975”.   Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт  Автоматически созданное описание |
| 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 loop join станет работать быстрее благодаря индексам на Н\_ЛЮДИ.ИД и Н\_СЕССИЯ.ЧЛВК\_ИД | |

1. **План выполнения 1 запроса**

|  |
| --- |
| Nested Loop (cost=4.58..19.63 rows=1 width=24) (actual time=0.047..0.048 rows=0 loops=1)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (actual time=0.035..0.036 rows=1 loops=1)  Index Cond: ("ИД" = 152862)  -> Bitmap Heap Scan on "Н\_СЕССИЯ" (cost=4.30..11.32 rows=1 width=8) (actual time=0.007..0.007 rows=0 loops=1)  Recheck Cond: ("ЧЛВК\_ИД" = 152862)  Filter: (("ИД" > 27640) AND ("ИД" < 1975))  -> Bitmap Index Scan on "SYS\_C003500\_IFK" (cost=0.00..4.29 rows=2 width=0) (actual time=0.005..0.005 rows=0 loops=1)  Index Cond: ("ЧЛВК\_ИД" = 152862)  Planning Time: 0.908 ms  Execution Time: 0.135 ms  Изображение выглядит как текст, диаграмма, линия, снимок экрана  Автоматически созданное описание |

1. **Уменьшение времени выполнения 2 запроса**

|  |  |
| --- | --- |
| Индексы, добавление которых уменьшит время выполнения запроса:  а) На таблице Н\_ЛЮДИ:   * Индекс на атрибуте ОТЧЕСТВО (B-tree). Это ускорит фильтрацию строк с условием “Н\_ЛЮДИ.ОТЧЕСТВО > ‘Владимирович’”.   b) На таблицах Н\_ОБУЧЕНИЯ и Н\_УЧЕНИКИ уже имеются индексы на столбцах ЧЛВК\_ИД.  Возможные планы выполнения запроса (без индексов): | |
| План 1:  Изображение выглядит как текст, диаграмма, снимок экрана, линия  Автоматически созданное описание | План 2:  Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт  Автоматически созданное описание |
| 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 loop join станет работать быстрее благодаря индексам на Н\_ОБУЧЕНИЯ.ИД и Н\_УЧЕНИКИ.ЧЛВК\_ИД | |

1. **План выполнения 2 запроса**

|  |
| --- |
| Nested Loop Left Join (cost=4.89..39.62 rows=5 width=14) (actual time=0.083..0.086 rows=1 loops=1)  Join Filter: ("Н\_ЛЮДИ"."ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД")  -> Nested Loop (cost=0.56..16.61 rows=1 width=10) (actual time=0.059..0.061 rows=1 loops=1)  -> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=4) (actual time=0.045..0.046 rows=1 loops=1)  Index Cond: ("ИД" = 163276)  Filter: (("ОТЧЕСТВО")::text > 'Владимирович'::text)  -> Index Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..8.30 rows=1 width=10) (actual time=0.011..0.012 rows=1 loops=1)  Index Cond: ("ЧЛВК\_ИД" = 163276)  -> Bitmap Heap Scan on "Н\_УЧЕНИКИ" (cost=4.33..22.95 rows=5 width=8) (actual time=0.019..0.020 rows=1 loops=1)  Recheck Cond: ("ЧЛВК\_ИД" = 163276)  Heap Blocks: exact=1  -> Bitmap Index Scan on "УЧЕН\_ОБУЧ\_FK\_I" (cost=0.00..4.32 rows=5 width=0) (actual time=0.016..0.016 rows=1 loops=1)  Index Cond: ("ЧЛВК\_ИД" = 163276)  Planning Time: 1.589 ms  Execution Time: 0.174 ms  Изображение выглядит как текст, диаграмма, снимок экрана, линия  Автоматически созданное описание |

## **Вывод:**

В ходе выполнения лабораторной работы я освоил основы реляционной алгебры, научился строить планы выполнения запросов. 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.