



**Вариант №7**  
**Лабораторная работа №4**  
**По дисциплине**  
**Базы Данных**

*Выполнил:*  
Студент группы Р3112  
Марьин Григорий Алексеевич

*Преподаватель:*  
Кустарев Иван Павлович

## 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.

Изменяются 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):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Ведомость.  
б) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД > 163249.  
в) 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)  
а) Н\_ЛЮДИ.ОТЧЕСТВО < Сергеевич.  
б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 163276.  
Вид соединения: RIGHT JOIN.

## 2. 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;
```

## 3. Уменьшение времени выполнения первого запроса

### Индексы, добавление которых уменьшит время выполнения запроса:

#### а) На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:

- Индекс на атрибуте НАИМЕНОВАНИЕ (B-tree) Это ускорит фильтрацию строк с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < 'Ведомость'".

#### б) 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:**

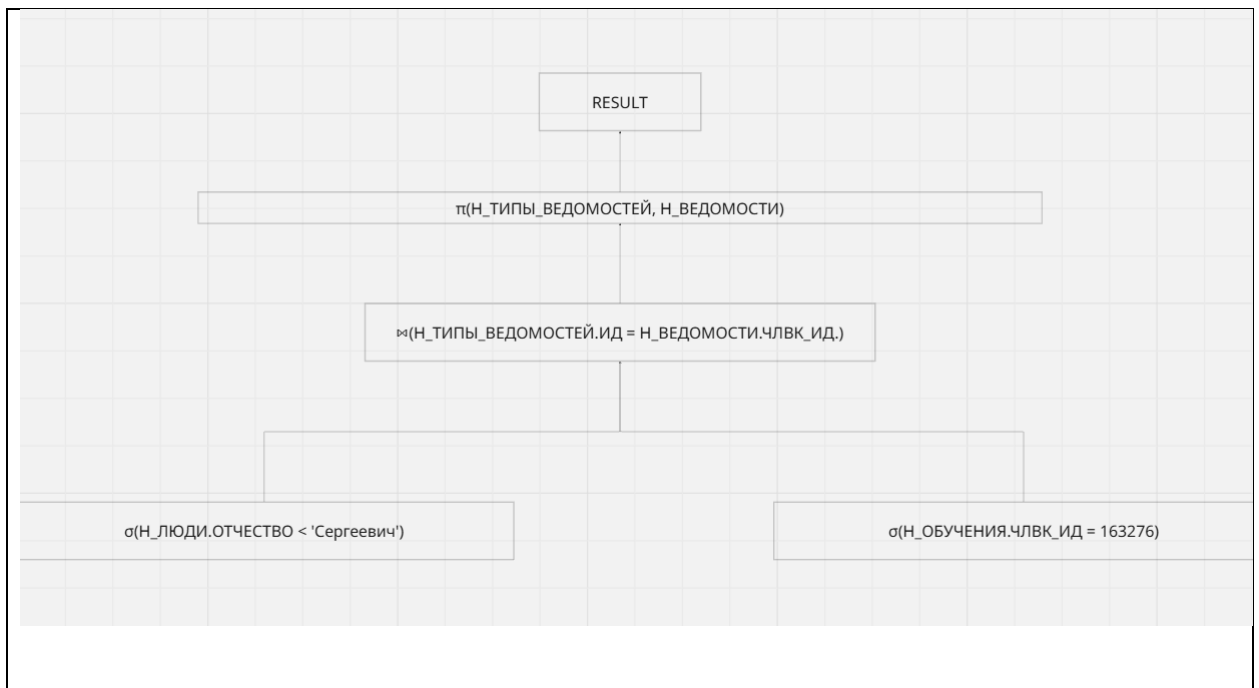
- План 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

**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.

## 4. 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;
```

## 2. Уменьшение времени выполнения 2 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

а) На таблице Н\_ЛЮДИ:

- Индекс на атрибуте ИМЯ (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ОТЧЕСТВО < 'Сергеевич'".

б) 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:

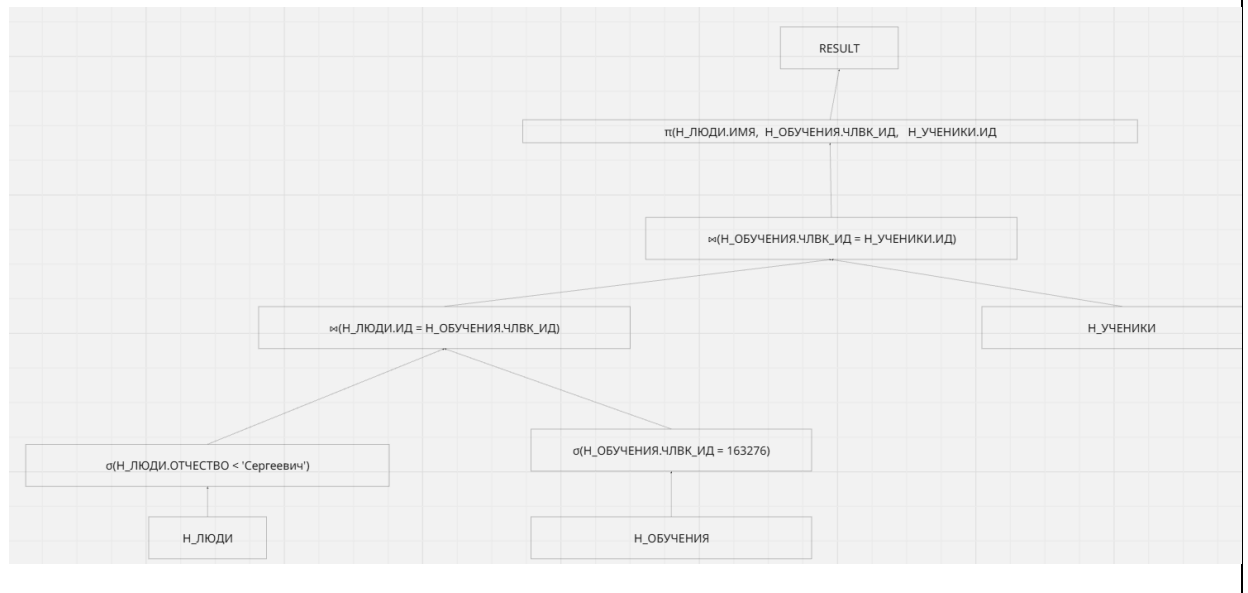
- Вместо полного скана таблиц будет использоваться индексный скан.
- Nested Loops Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД.

## 3. 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_ПК" 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_ПК" 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



#### 4. 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.