

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»**



ITMO UNIVERSITY

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4
по дисциплине
«БАЗЫ ДАННЫХ»:

Вариант 1373

Выполнил:

Ястребов-Амирханов Алекси

ГРУППА: Р3132

ПРЕПОДАВАТЕЛЬ: Афанасьев Дмитрий Борисович

Санкт-Петербург,
2024

Оглавление

Задание	3
Выполнение	4
Реализация запросов на SQL.....	4
Уменьшение времени выполнения 1 запроса	5
Уменьшение времени выполнения 2 запроса	7
Вывод	9

Задание

Составить запросы на языке SQL (пункты 1-2).

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).

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):

а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.

б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 142390.

с) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 163249.

Вид соединения: LEFT JOIN.

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.ГРУППА.

Фильтры: (AND)

а) Н_ЛЮДИ.ФАМИЛИЯ < Соколов.

б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276.

Вид соединения: INNER JOIN.

Выполнение

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):
-- а) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Экзаменационный лист.
-- б) Н_ВЕДОМОСТИ.ЧЛВК_ИД > 142390.
-- с) Н_ВЕДОМОСТИ.ЧЛВК_ИД = 163249.
-- Вид соединения: LEFT JOIN.
```

SELECT

Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД,
Н_ВЕДОМОСТИ.ДАТА

FROM

Н_ТИПЫ_ВЕДОМОСТЕЙ

LEFT JOIN

Н_ВЕДОМОСТИ ON Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = Н_ВЕДОМОСТИ.ТВ_ИД

WHERE

Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'
AND Н_ВЕДОМОСТИ.ЧЛВК_ИД > 142390
AND Н_ВЕДОМОСТИ.ЧЛВК_ИД = 163249;

```
-- 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.ГРУППА.
-- Фильтры: (AND)
-- а) Н_ЛЮДИ.ФАМИЛИЯ < Соколов.
-- б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276.
-- Вид соединения: INNER JOIN.
```

SELECT

Н_ЛЮДИ.ИМЯ,
Н_ОБУЧЕНИЯ.ЧЛВК_ИД,
Н_УЧЕНИКИ.ГРУППА

FROM

Н_ЛЮДИ

INNER JOIN

Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД

INNER JOIN

Н_УЧЕНИКИ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ЧЛВК_ИД

WHERE

Н_ЛЮДИ.ФАМИЛИЯ < 'Соколов'
AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276;

Уменьшение времени выполнения 1 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

1) Таблица Н_ТИПЫ_ВЕДОМОСТЕЙ

Атрибут: НАИМЕНОВАНИЕ

Причина: Ускоряет поиск строк по условию НАИМЕНОВАНИЕ = 'Экзаменационный лист'.

```
-- CREATE INDEX idx_тип_ведомостей_наименование ON  
Н_ТИПЫ_ВЕДОМОСТЕЙ(НАИМЕНОВАНИЕ);
```

2) Таблица Н_ВЕДОМОСТИ

Атрибут: ЧЛВК_ИД

Причина: Ускоряет фильтрацию строк по условиям ЧЛВК_ИД > 142390 и ЧЛВК_ИД = 163249.

```
-- CREATE INDEX idx_ведомости_члвк_ид ON Н_ВЕДОМОСТИ(ЧЛВК_ИД);
```

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запроса без индексов:

План 1:

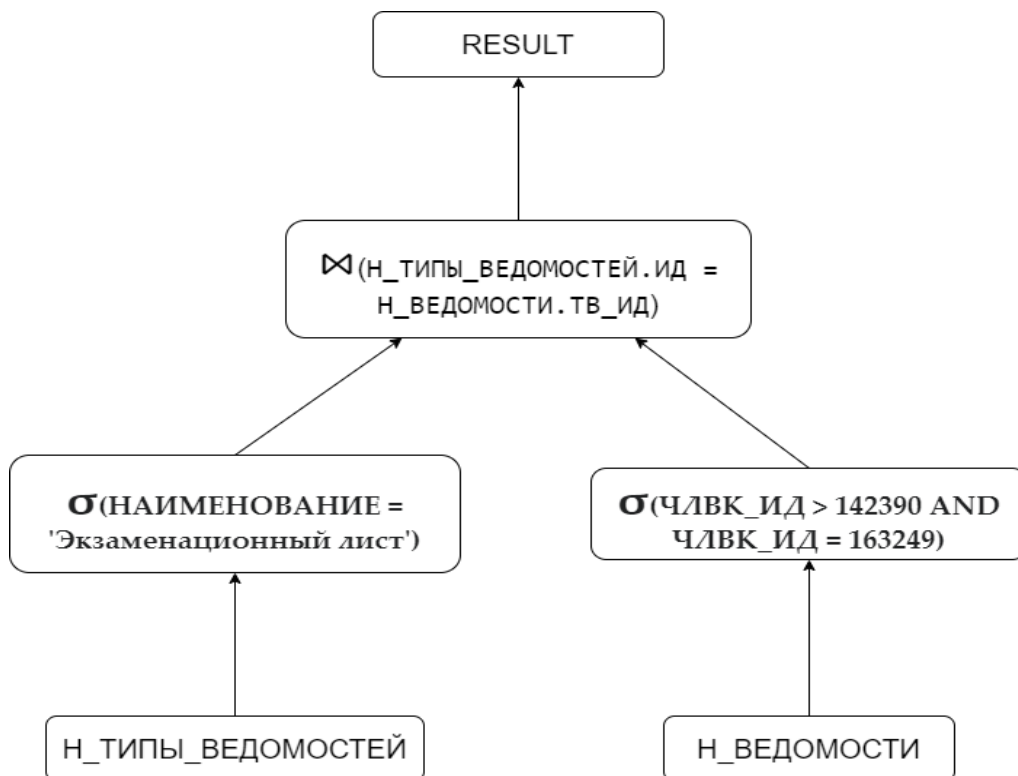
- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ.
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию Н_ВЕДОМОСТИ.ЧЛВК_ИД > 142390 и Н_ВЕДОМОСТИ.ЧЛВК_ИД = 163249.
- Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ_ИД.
- Фильтрация результата соединения по условию Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'.

План 2:

- Полный скан таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ с применением фильтра по условию Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'.
- Полный скан таблицы Н_ВЕДОМОСТИ с применением фильтра по условию Н_ВЕДОМОСТИ.ЧЛВК_ИД > 142390 и 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.
- Индексный скан таблицы H_ТИПЫ_ВЕДОМОСТЕЙ по индексу на атрибуте НАИМЕНОВАНИЕ для условия H_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = 'Экзаменационный лист'.
- Индексный скан таблицы H_ВЕДОМОСТИ по индексу на атрибуте ЧЛВК_ИД для условий H_ВЕДОМОСТИ.ЧЛВК_ИД > 142390 и H_ВЕДОМОСТИ.ЧЛВК_ИД = 163249.
- Соединение таблиц с использованием Nested Loops Join или Hash Join, ускоренное индексом на атрибуте ТВ_ИД.

QUERY PLAN

```
Nested Loop (cost=0.29..45.32 rows=4 width=12) (actual time=0.026..0.038 rows=1 loops=1)
  Join Filter: ("H_ТИПЫ_ВЕДОМОСТЕЙ"."ИД" = "H_ВЕДОМОСТИ"."ТВ_ИД")
  Rows Removed by Join Filter: 18
  -> Seq Scan on "H_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual
time=0.012..0.013 rows=1 loops=1)
    Filter: (("НАИМЕНОВАНИЕ")::text = 'Экзаменационный лист'::text)
    Rows Removed by Filter: 2
  -> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "H_ВЕДОМОСТИ" (cost=0.29..44.12 rows=13
width=12) (actual time=0.010..0.017 rows=19 loops=1)
    Index Cond: (("ЧЛВК_ИД" > 142390) AND ("ЧЛВК_ИД" = 163249))
Planning Time: 0.313 ms
Execution Time: 0.078 ms
(10 строк)
```

Уменьшение времени выполнения 2 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

1) Таблица Н_ЛЮДИ

Атрибут: ФАМИЛИЯ

Причина: Ускоряет фильтрацию строк по условию ФАМИЛИЯ < 'Соколов'.

```
-- CREATE INDEX idx_люди_фамилия ON Н_ЛЮДИ(ФАМИЛИЯ);
```

2) Таблица Н_ОБУЧЕНИЯ

Атрибут: ЧЛВК_ИД

Причина: Улучшит производительность фильтрации по условию

Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276 и соединения с таблицей Н_ЛЮДИ по атрибуту ЧЛВК_ИД.

```
-- CREATE INDEX idx_обучения_члвк_ид ON Н_ОБУЧЕНИЯ(ЧЛВК_ИД);
```

Добавление индекса позволит ускорить поиск нужных записей по фильтрам в запросе.

Возможные планы выполнения запроса без индексов:

План 1:

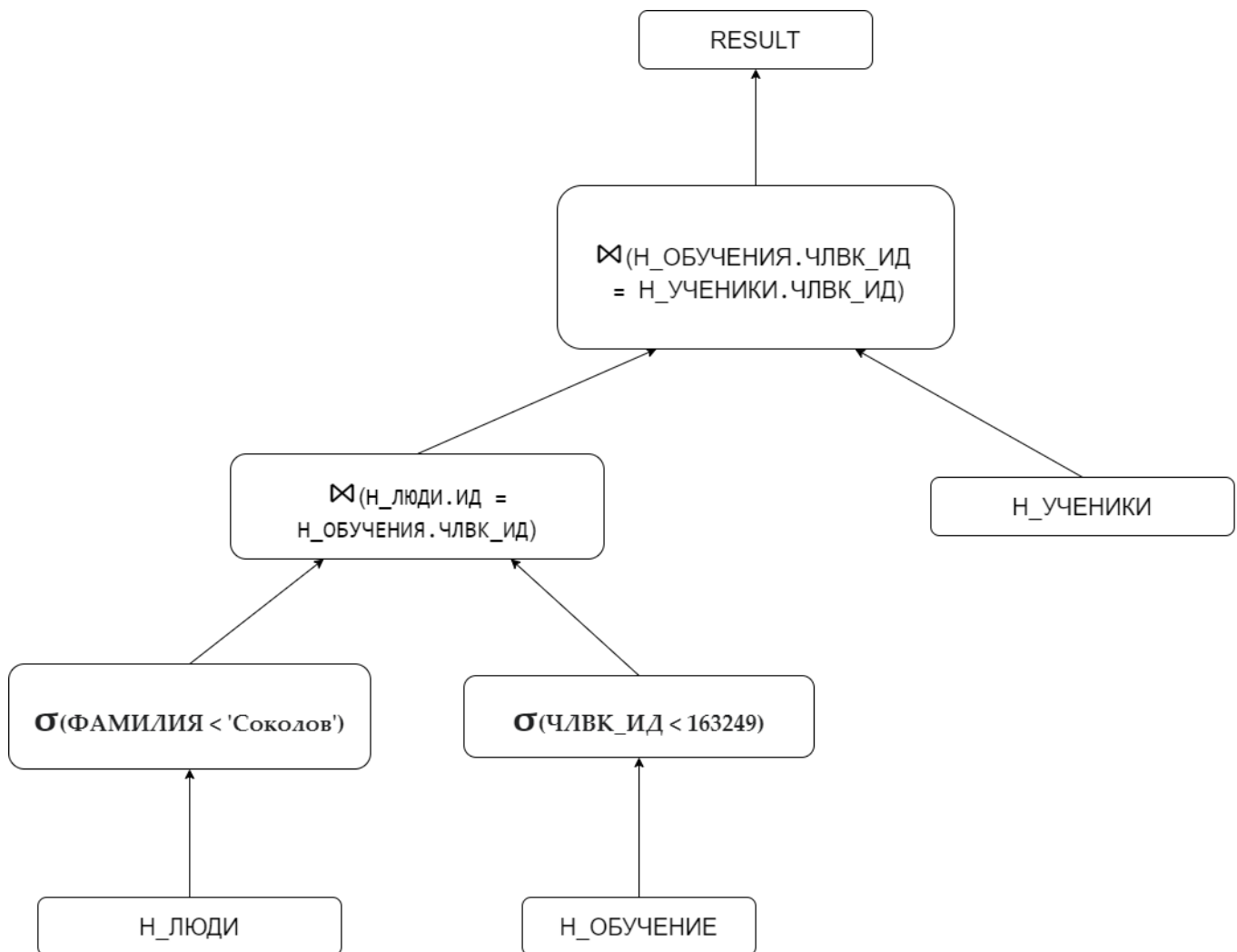
- Полный скан таблицы Н_ЛЮДИ с применением фильтра по условию Н_ЛЮДИ.ФАМИЛИЯ < 'Соколов'.
- Полный скан таблицы Н_ОБУЧЕНИЯ с применением фильтра по условию Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276.
- Полный скан таблицы Н_УЧЕНИКИ.
- Соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД.
- Соединение таблиц Н_ОБУЧЕНИЯ и Н_УЧЕНИКИ с использованием Nested Loops Join по атрибуту ЧЛВК_ИД.

План 2:

- Полный скан таблицы Н_ЛЮДИ.
- Полный скан таблицы Н_ОБУЧЕНИЯ.
- Полный скан таблицы Н_УЧЕНИКИ.
- Соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ с использованием Hash Join по атрибуту ЧЛВК_ИД.
- Применение фильтров по условиям Н_ЛЮДИ.ФАМИЛИЯ < 'Соколов' и Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276.
- Соединение результирующего набора с таблицей Н_УЧЕНИКИ с использованием Hash Join по атрибуту ЧЛВК_ИД.

Оптимальный план:

План 1 является оптимальным, так как он выполняет фильтрацию данных до соединения таблиц, что уменьшает количество строк для обработки и ускоряет выполнение запроса.



При добавлении индексов план выполнения запроса изменился :

- Индексный скан таблицы **Н_ЛЮДИ** по атрибуту **ФАМИЛИЯ** для условия $\text{H_ЛЮДИ.ФАМИЛИЯ} < \text{'Соколов'}$.
- Индексный скан таблицы **Н_ОБУЧЕНИЯ** по атрибуту **ЧЛВК_ИД** для условия $\text{H_ОБУЧЕНИЯ.ЧЛВК_ИД} < 163276$.
- Индексный скан таблицы **Н_УЧЕНИКИ** по атрибуту **ЧЛВК_ИД**.
- Соединение таблиц **Н_ЛЮДИ** и **Н_ОБУЧЕНИЯ** с использованием Nested Loops Join по атрибуту **ЧЛВК_ИД**.
- Соединение таблиц **Н_ОБУЧЕНИЯ** и **Н_УЧЕНИКИ** с использованием Nested Loops Join по атрибуту **ЧЛВК_ИД**.

После добавления индексов, **План 1** остаётся оптимальным, так как фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки и ускоряет выполнение запроса. Использование индексных сканов вместо полного сканирования таблиц значительно уменьшает количество обрабатываемых строк, что ускоряет выполнение запроса.

QUERY PLAN

```
Hash Join (cost=786.24..1826.85 rows=17909 width=21) (actual time=7.142..18.510 rows=18508 loops=1)
  Hash Cond: ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
    -> Seq Scan on "Н_УЧЕНИКИ" (cost=0.00..774.11 rows=23311 width=8) (actual time=0.007..3.974 rows=23311 loops=1)
    -> Hash (cost=737.09..737.09 rows=3932 width=21) (actual time=7.102..7.104 rows=3929 loops=1)
      Buckets: 4096 Batches: 1 Memory Usage: 244kB
      -> Hash Join (cost=604.15..737.09 rows=3932 width=21) (actual time=4.280..6.296 rows=3929 loops=1)
        Hash Cond: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
          -> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=5014 width=4) (actual time=0.014..0.886 rows=5019 loops=1)
            Filter: ("ЧЛВК_ИД" < 163276)
            Rows Removed by Filter: 2
          -> Hash (cost=553.98..553.98 rows=4014 width=17) (actual time=4.234..4.235 rows=4014 loops=1)
            Buckets: 4096 Batches: 1 Memory Usage: 233kB
            -> Seq Scan on "Н_ЛЮДИ" (cost=0.00..553.98 rows=4014 width=17) (actual time=0.035..3.465 rows=4014 loops=1)
              Filter: (("ФАМИЛИЯ")::text < 'Соколов'::text)
              Rows Removed by Filter: 1104
Planning Time: 2.596 ms
Execution Time: 19.511 ms
(17 строк)
```

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.