Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4
Вариант 123

Выполнил:
Лежнев Никита Сергеевич
Группа P3112
Проверил:
Кустарев Иван Павлович

Санкт-Петербург 2025г.

Содержание:

Задание 3

Реализация запроса на SQL 4

Уменьшение времени выполнения 1 запроса 4

Возможные планы выполнения запроса 1: 5

План выполнения запроса 1 6

Уменьшение времени выполнения 2 запроса 7

Возможные планы выполнения запроса 2: 7

План выполнения запроса 2: 8

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.
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):
   a) Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Перезачет.
   b) Н_ВЕДОМОСТИ.ДАТА = 2010-06-18.
   Вид соединения: INNER 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.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.
   Фильтры: (AND)
   a) Н_ЛЮДИ.ОТЧЕСТВО > Сергеевич.
   b) Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 105590.
   c) Н_УЧЕНИКИ.ИД < 150308.
   Вид соединения: RIGHT JOIN.

Реализация запроса на SQL

```
--- Запрос 1 ----
SELECT
   tv.ИД AS ТипВедомости_ИД,
   v.ДАТА AS Ведомость_Дата
FROM
   H_ТИПЫ_ВЕДОМОСТЕЙ tv
INNER JOIN
   H_ВЕДОМОСТИ v ON tv.ИД = v.ТВ_ИД
WHERE
   tv.НАИМЕНОВАНИЕ = 'Перезачет'
   AND v.ДАТА = '2010-06-18';

--- Запрос 2 ----
SELECT
   l.ИД AS Люди_ИД,
   o.ЧЛВК_ИД AS Обучения_ЧЛВК_ИД,
   u.ИД AS Ученики_ИД
FROM
   H_ЛЮДИ l
INNER JOIN
   H_ОБУЧЕНИЯ o ON l.ИД = o.ЧЛВК_ИД
RIGHT JOIN
   H_УЧЕНИКИ u ON o.ЧЛВК_ИД = u.ЧЛВК_ИД
WHERE
   l.ОТЧЕСТВО > 'Сергеевич'
   AND o.ЧЛВК_ИД > 105590
   AND u.ИД < 150308;
```

Уменьшение времени выполнения 1 запроса

1. Таблица: H_ТИПЫ_ВЕДОМОСТЕЙ
    - Атрибуты для индекса: (НАИМЕНОВАНИЕ, ИД)
    - Тип индекса: B-tree
    - Объяснение:
        - Атрибут НАИМЕНОВАНИЕ используется в условии WHERE tv.НАИМЕНОВАНИЕ = 'Перезачет'. Индекс по этому полю позволит быстро найти строки с указанным значением НАИМЕНОВАНИЕ, избегая полного сканирования таблицы.
        - Включение атрибута ИД в этот же композитный индекс делает его покрывающим для таблицы H_ТИПЫ_ВЕДОМОСТЕЙ в данном запросе. Атрибут ИД используется в SELECT (tv.ИД) и в условии соединения (ON tv.ИД = v.ТВ_ИД). Таким образом, СУБД сможет получить оба значения (НАИМЕНОВАНИЕ для фильтра и ИД для соединения и вывода) непосредственно из индекса, не обращаясь к самой таблице.

2. Таблица: Н_ВЕДОМОСТИ
    - Атрибуты для индекса: (ДАТА, ТВ_ИД)
    - Тип индекса: B-tree
    - Объяснение:
        - Атрибут ДАТА используется в условии WHERE v.ДАТА = '2010-06-18'. Индекс по этому полю позволит быстро найти строки с указанной датой.
        - Включение атрибута ТВ_ИД в этот же композитный индекс помогает эффективно выполнить соединение (ON tv.ИД = v.ТВ_ИД) после фильтрации по ДАТА. Атрибут ДАТА также используется в SELECT (v.ДАТА). Этот индекс будет частично покрывающим, так как для SELECT и JOIN из таблицы Н_ВЕДОМОСТИ нужны только ДАТА и ТВ_ИД.
        - Можно сделать иначе: (ТВ_ИД, ДАТА). Если СУБД сначала отфильтрует Н_ТИПЫ_ВЕДОМОСТЕЙ и получит небольшой набор tv.ИД, то для каждой такой tv.ИД ей потребуется быстро найти соответствующие v.ТВ_ИД в Н_ВЕДОМОСТИ и проверить условие по ДАТА. В этом случае порядок (ТВ_ИД, ДАТА) может быть предпочтительнее. Оптимизатор сам выберет наилучший путь.

Возможные планы выполнения запроса 1:

A. План A: Nested Loop Join (Н_ТИПЫ_ВЕДОМОСТЕЙ)
   Полное сканирование (Sequential Scan) таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ.
   Для каждой строки применить фильтр tv.НАИМЕНОВАНИЕ = 'Перезачет'.
   Если строка подходит, для её tv.ИД выполнить полное сканирование (Sequential Scan) таблицы Н_ВЕДОМОСТИ.
   Для каждой строки из Н_ВЕДОМОСТИ проверить условие соединения tv.ИД = v.ТВ_ИД.
   Применить фильтр v.ДАТА = '2010-06-18'.
   Если все условия выполнены, вернуть выбранные атрибуты.
   Недостатки: Очень неэффективно (O(N*M)) если обе таблицы большие.

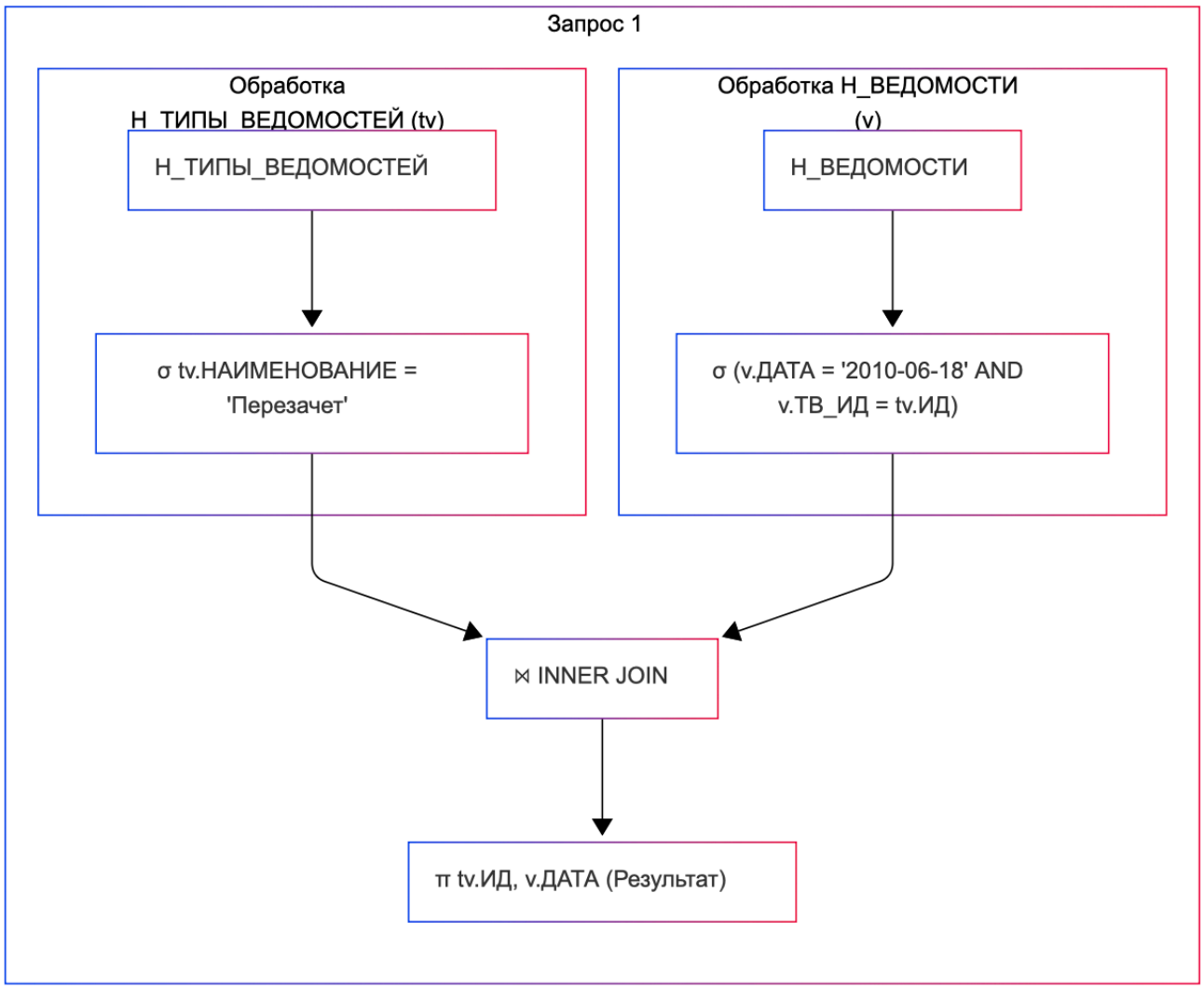
B. План B: Hash Join
   Полное сканирование (Sequential Scan) таблицы Н_ТИПЫ_ВЕДОМОСТЕЙ.
   Применить фильтр tv.НАИМЕНОВАНИЕ = 'Перезачет'. Отобранные строки (или только tv.ИД) поместить в хеш-таблицу по ключу tv.ИД.
   Полное сканирование (Sequential Scan) таблицы Н_ВЕДОМОСТИ. Применить фильтр v.ДАТА = '2010-06-18'.
   Для каждой отфильтрованной строки из Н_ВЕДОМОСТИ проверить наличие v.ТВ_ИД в хеш-таблице.
   Если найдено совпадение, вернуть выбранные атрибуты.
   Преимущества: Обычно эффективнее Nested Loop Join для больших таблиц без индексов, если хеш-таблица помещается в память.

Оптимальный план (без индексов):
План B (Hash Join), так как он обычно выполняет меньше операций ввода-вывода по сравнению с Nested Loop Join на больших неиндексированных таблицах. Фильтрация до или во время построения/пробирования хеш-таблицы уменьшает объем обрабатываемых данных.

План выполнения запроса 1

QUERY PLAN

Nested Loop (cost=0.29..218.77 rows=24 width=12) (actual time=0.105..0.106 rows=0 loops=1)
  Join Filter: (tv."ИД" = v."ТВ_ИД")
  Rows Removed by Join Filter: 141
  -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" tv (cost=0.00..1.04 rows=1 width=4) (actual time=0.017..0.018 rows=1 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text = 'Перезачет'::text)
        Rows Removed by Filter: 2
  -> Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" v (cost=0.29..216.82 rows=73 width=12) (actual time=0.016..0.067 rows=141 loops=1)
        Index Cond: ("ДАТА" = '2010-06-18 00:00:00'::timestamp without time zone)
Planning Time: 0.272 ms
Execution Time: 0.139 ms



Уменьшение времени выполнения 2 запроса

1. Таблица: Н_ЛЮДИ
   - Атрибуты для индекса: (ОТЧЕСТВО, ИД)
   - Тип индекса: B-tree
   - Объяснение:
     - Атрибут ОТЧЕСТВО используется в WHERE l.ОТЧЕСТВО > 'Сергеевич' (диапазонный поиск). Индекс позволит быстро отфильтровать строки.
     - Включение ИД (используется в SELECT l.ИД и в JOIN ON l.ИД = o.ЧЛВК_ИД) делает индекс покрывающим для этой таблицы.
2. Таблица: Н_ОБУЧЕНИЯ
   - Атрибуты для индекса: (ЧЛВК_ИД)
   - Тип индекса: B-tree
   - Объяснение:
     - Атрибут ЧЛВК_ИД используется очень активно: в WHERE o.ЧЛВК_ИД > 105590 (диапазонный поиск), в JOIN ON l.ИД = o.ЧЛВК_ИД, в JOIN ON o.ЧЛВК_ИД = u.ЧЛВК_ИД и в SELECT o.ЧЛВК_ИД. Индекс по этому полю очень важен для всех этих операций.
3. Таблица: Н_УЧЕНИКИ
   - Атрибуты для индекса: (ИД, ЧЛВК_ИД)
   - Тип индекса: B-tree
   - Объяснение:
     - Атрибут ИД используется в WHERE u.ИД < 150308 (диапазонный поиск). Индекс позволит быстро отфильтровать строки.
     - Включение ЧЛВК_ИД (используется в JOIN ON o.ЧЛВК_ИД = u.ЧЛВК_ИД) и использование ИД в SELECT u.ИД делает этот индекс покрывающим для данной таблицы. Порядок (ИД, ЧЛВК_ИД) лучше, так как фильтрация по диапазону на ИД будет первой операцией.

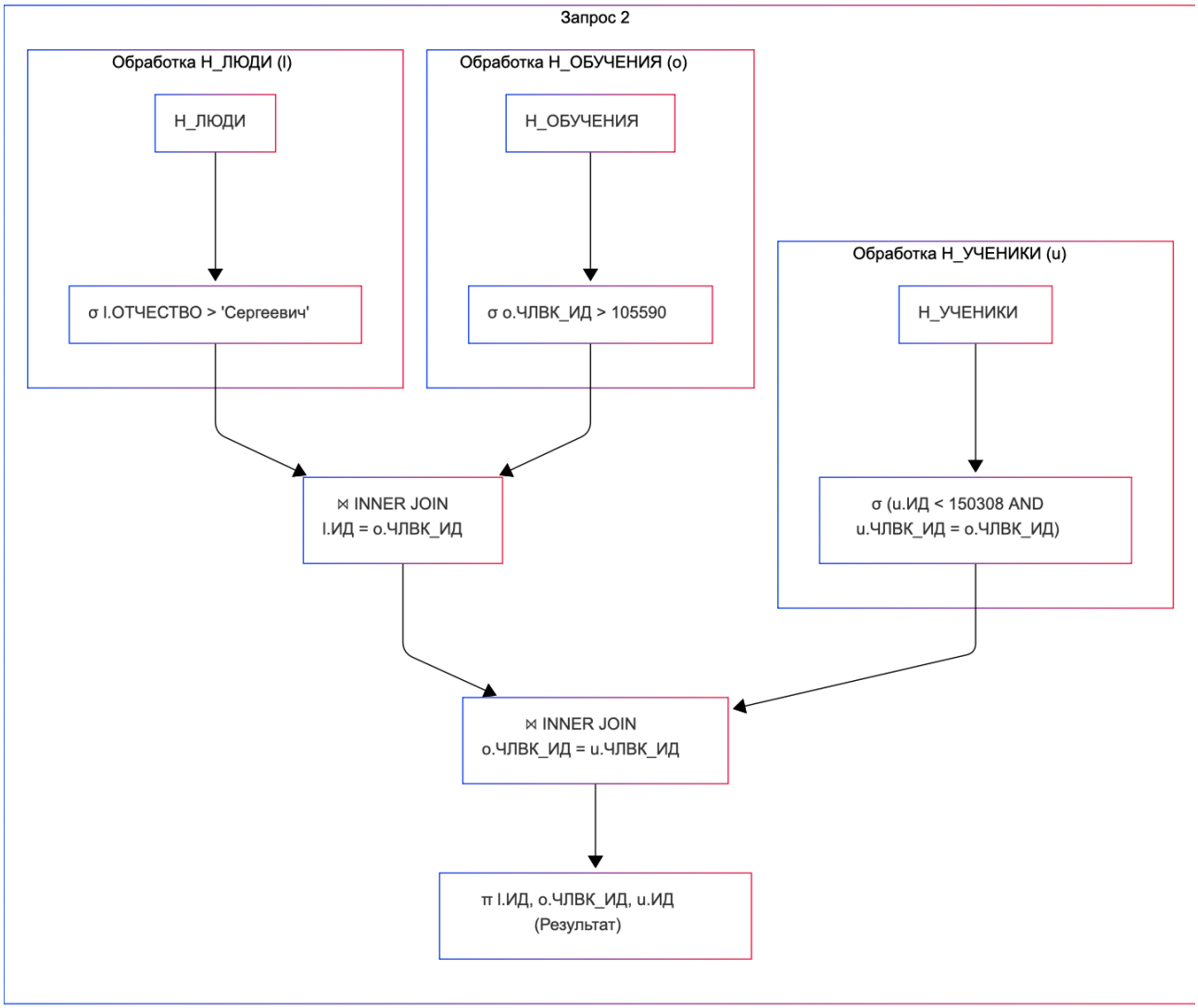
Возможные планы выполнения запроса 2:

A. План A: Последовательные Nested Loop Joins
   Полное сканирование Н_ЛЮДИ (l), фильтр l.ОТЧЕСТВО > 'Сергеевич'.
   Для каждой подходящей строки l: полное сканирование Н_ОБУЧЕНИЯ (o), фильтр o.ЧЛВК_ИД > 105590, проверка l.ИД = o.ЧЛВК_ИД.
   Для каждой подходящей пары (l,o): полное сканирование Н_УЧЕНИКИ (u), фильтр u.ИД < 150308, проверка o.ЧЛВК_ИД = u.ЧЛВК_ИД.
   Очень неэффективно для сколько-нибудь больших таблиц.
B. План B: Последовательные Hash Joins
   Полное сканирование Н_ЛЮДИ (l), фильтр l.ОТЧЕСТВО > 'Сергеевич'. Результат L'.
   Полное сканирование Н_ОБУЧЕНИЯ (o), фильтр o.ЧЛВК_ИД > 105590. Результат O'.
   Соединить L' и O' через Hash Join по l.ИД = o.ЧЛВК_ИД. Результат LO'.
   Полное сканирование Н_УЧЕНИКИ (u), фильтр u.ИД < 150308. Результат U'.
   Соединить LO' и U' через Hash Join по o.ЧЛВК_ИД = u.ЧЛВК_ИД. (Этот JOIN будет фактически INNER из-за условий WHERE).

Оптимальный план (без индексов):
План B (Последовательные Hash Joins). Он более эффективен для больших таблиц, так как каждое соединение выполняется за линейное время от суммарного размера соединяемых (уже отфильтрованных) наборов данных.

План выполнения запроса 2:

```
QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
Nested Loop  (cost=169.64..687.44 rows=1352 width=12) (actual time=2.856..6.297 rows=1285 loops=1)
  Join Filter: (l."ИД" = u."ЧЛВК_ИД")
  ->  Hash Join  (cost=169.35..302.30 rows=422 width=8) (actual time=2.827..4.091 rows=426 loops=1)
        Hash Cond: (o."ЧЛВК_ИД" = l."ИД")
        ->  Seq Scan on "H_ОБУЧЕНИЯ" o  (cost=0.00..119.76 rows=5020 width=4) (actual time=0.013..0.734 rows=5020 loops=1)
              Filter: ("ЧЛВК_ИД" > 105590)
              Rows Removed by Filter: 1
        ->  Hash  (cost=163.97..163.97 rows=430 width=4) (actual time=2.788..2.789 rows=430 loops=1)
              Buckets: 1024  Batches: 1  Memory Usage: 24kB
              ->  Seq Scan on "H_ЛЮДИ" l  (cost=0.00..163.97 rows=430 width=4) (actual time=0.006..2.700 rows=430 loops=1)
                    Filter: (("ОТЧЕСТВО")::text > 'Сергеевич'::text)
                    Rows Removed by Filter: 4688
  ->  Index Scan using "УЧЕН_ОБУЧ_FK_I" on "H_УЧЕНИКИ" u  (cost=0.29..0.88 rows=3 width=8) (actual time=0.002..0.004 rows=3 loops=426)
        Index Cond: ("ЧЛВК_ИД" = o."ЧЛВК_ИД")
        Filter: ("ИД" < 150308)
        Rows Removed by Filter: 1
 Planning Time: 1.502 ms
 Execution Time: 6.416 ms
```


Запрос 2
Обработка Н_ЛЮДИ (l)
Н_ЛЮДИ
σ l.ОТЧЕСТВО > 'Сергеевич'
Обработка Н_ОБУЧЕНИЯ (o)
Н_ОБУЧЕНИЯ
σ o.ЧЛВК_ИД > 105590
Обработка Н_УЧЕНИКИ (u)
Н_УЧЕНИКИ
σ (u.ИД < 150308 AND
u.ЧЛВК_ИД = o.ЧЛВК_ИД)
⋈ INNER JOIN
l.ИД = o.ЧЛВК_ИД
⋈ INNER JOIN
o.ЧЛВК_ИД = u.ЧЛВК_ИД
π l.ИД, o.ЧЛВК_ИД, u.ИД
(Результат)