

Федеральное государственное автономное образовательное учреждение
высшего образования “Национальный исследовательский университет ИТМО”

Факультет Программной Инженерии И Компьютерной Техники

Лабораторная работа №4

Вариант 464900

Выполнила:

Абдуллаева София Улугбековна

Группа Р3108

Проверила:

Заболотняя Ольга Михайловна

Санкт-Петербург 2025

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).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ЛЮДИ.ИД < 100865.
б) Н_ВЕДОМОСТИ.ДАТА = 1998-01-05.
Вид соединения: RIGHT 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)
а) Н_ЛЮДИ.ИМЯ < Роман.
б) Н_ОБУЧЕНИЯ.НЗК < 001000.
с) Н_УЧЕНИКИ.ГРУППА < 4100.
Вид соединения: RIGHT JOIN.

Запрос №1

Реализация запроса

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ЛЮДИ.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ИД.

Фильтры (AND):

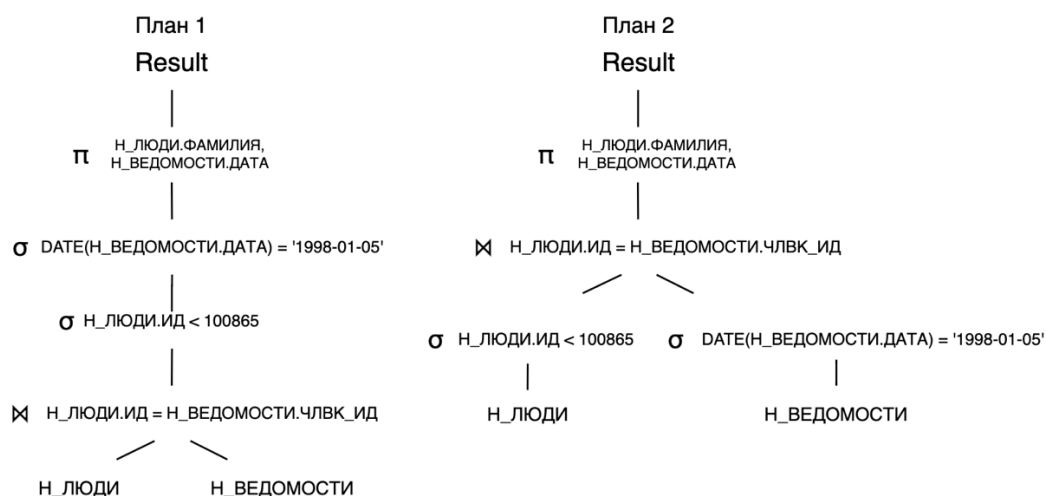
а) Н_ЛЮДИ.ИД < 100865.

б) Н_ВЕДОМОСТИ.ДАТА = 1998-01-05.

Вид соединения: RIGHT JOIN.

```
SELECT "Н_люди"."ФАМИЛИЯ", "Н_ведомости"."ИД"  
FROM "Н_люди"  
      RIGHT JOIN "Н_ведомости" ON "Н_люди"."ИД" = "Н_ведомости"."ЧЛВК_ИД"  
WHERE "Н_люди"."ИД" < 100865  
      AND DATE("Н_ведомости"."ДАТА") = '1998-01-05';
```

Планы выполнения запросов



Описание плана №1: происходит полный скан таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ, соединение таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ по условию Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД, дальше фильтрация данных по 2 условиям Н_ЛЮДИ.ИД < 100865 и DATE(Н_ВЕДОМОСТИ.ДАТА) = '1998-01-05' и выбор атрибутов: Н_ЛЮДИ.ФАМИЛИЯ и Н_ВЕДОМОСТИ.ИД

В итоге происходит обработка всех строк в том числе тех, которые потом отсеиваются после выборки данных, поэтому этот подход будет медленнее и менее эффективным для таблиц с большим кол-вом строк.

Описание плана №2: происходит полный скан таблицы Н_ЛЮДИ и фильтрация строк по условию Н_ЛЮДИ.ИД < 100865, дальше полный скан таблицы Н_ВЕДОМОСТИ и фильтрация строк по условию DATE(Н_ВЕДОМОСТИ.ДАТА)='1998-01-05', теперь соединение таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ по условию Н_ЛЮДИ.ИД=Н_ВЕДОМОСТИ.ЧЛВК_ИД и выбор атрибутов: Н_ЛЮДИ.ФАМИЛИЯ и Н_ВЕДОМОСТИ.ИД

В данном подходе фильтрация строк происходит до соединения таблиц, получается, само соединение происходит по уже выбранным атрибутам, а не по таблицам целиком

Вывод: план №2 будет более эффективным по сравнению с планом №1

Применение индексов

```
CREATE INDEX statement_person_id_ind ON "Н_ВЕДОМОСТИ" USING hash("ЧЛВК_ИД");
CREATE INDEX statement_date_ind ON "Н_ВЕДОМОСТИ" USING hash("ДАТА");
CREATE INDEX person_id_ind ON "Н_ЛЮДИ" USING btree("ИД");
```

Добавление индексов на Н_ВЕДОМОСТИ.ЧЛВК_ИД и на Н_ЛЮДИ.ИД ускорит соединение таблиц Н_ЛЮДИ и Н_ВЕДОМОСТИ: для Н_ВЕДОМОСТИ.ЧЛВК_ИД был выбран hash-индекс, он использует хэш-функцию для быстрого поиска и применяется при операторе равенства, для Н_ЛЮДИ.ИД был выбран индекс b-tree, он тоже подходит для оператора равенства, в итоге быстро находятся строки для объединения обеих таблиц. Также индекс b-tree используется для диапазонных запросов, поэтому он подойдет для случая с фильтрацией WHERE Н_ЛЮДИ.ИД < 100865

Изменения планов при добавлении индексов

Планы выполнения запросов изменятся при добавлении индексов, будет выполняться индексный скан таблицы вместо полного скана

Вывод команды EXPLAIN ANALYZE

```
QUERY PLAN
Nested Loop  (cost=0.58..4602.21 rows=6 width=20) (actual time=0.091..0.091 rows=0 loops=1)
->  Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ"  (cost=0.28..100.71 rows=26 width=20) (actual time=0.006..0.036 rows=40 ...)
      Index Cond: ("ИД" < 100865)
->  Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ"  (cost=0.29..173.12 rows=1 width=8) (actual time=0.001..0.001 rows=0 loops=1)
      Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
      Filter: (date("ДАТА") = '1998-01-05'::date)
Planning Time: 0.254 ms
Execution Time: 0.113 ms
```

Запрос №2

Реализация запроса

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)

а) Н_ЛЮДИ.ИМЯ < Роман.

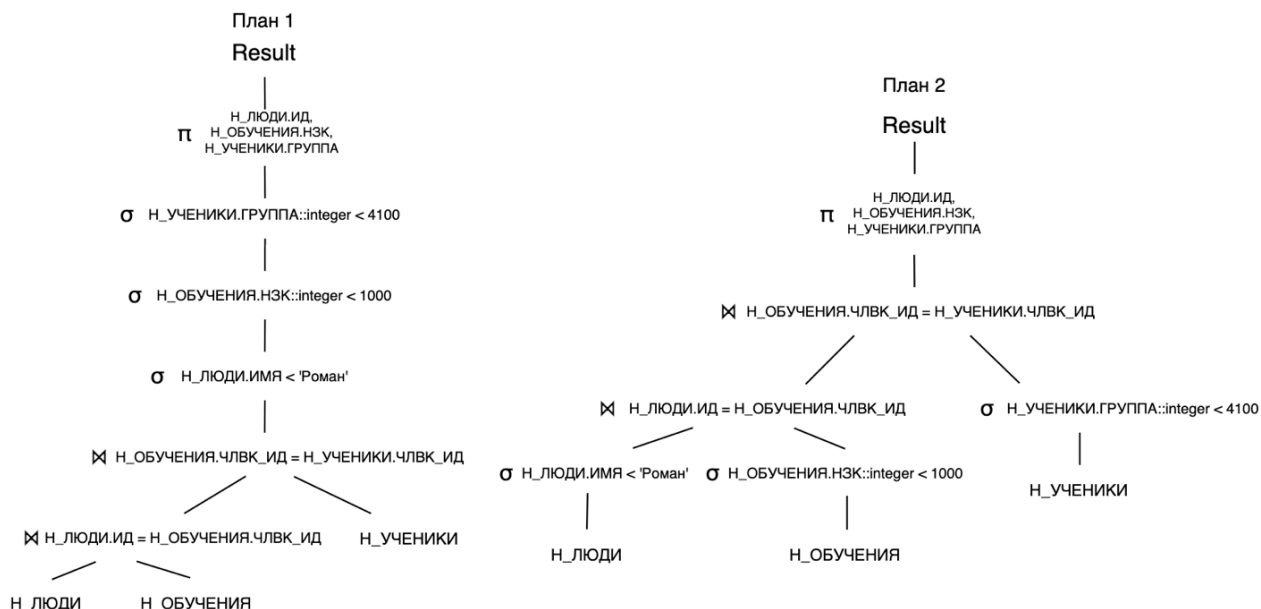
б) Н_ОБУЧЕНИЯ.НЗК < 001000.

с) Н_УЧЕНИКИ.ГРУППА < 4100.

Вид соединения: RIGHT JOIN.

```
SELECT "Н_люди"."ИД", "Н_обучения"."НЗК", "Н_ученики"."ГРУППА"
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"."ЧЛВК_ИД"
WHERE "Н_люди"."ИМЯ" < 'Роман' AND "Н_обучения"."НЗК"::integer < 1000 AND
      "Н_ученики"."ГРУППА"::integer < 4100;
```

Планы выполнения запросов



Описание плана №1: происходит полный скан таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ, соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ по условию Н_ЛЮДИ.ИД=Н_ОБУЧЕНИЯ.ЧЛВК_ИД, дальше полный скан таблицы Н_УЧЕНИКИ, соединение таблиц Н_ОБУЧЕНИЯ и Н_УЧЕНИКИ по условию Н_ОБУЧЕНИЯ.ЧЛВК_ИД=

Н_УЧЕНИКИ.ЧЛВК_ИД. Далее фильтрация данных по 3 условиям
Н_ЛЮДИ.ИМЯ < 'Роман', Н_ОБУЧЕНИЯ.НЗК::integer < 1000 и
Н_УЧЕНИКИ.ГРУППА::integer < 4100 и выбор атрибутов: Н_ЛЮДИ.ИД,
Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА

Происходит обработка всех строк в том числе тех, которые потом отсеиваются после выборки данных, поэтому этот подход будет медленнее и менее эффективным для таблиц с большим кол-вом строк

Описание плана №2: происходит полный скан таблицы Н_ОБУЧЕНИЯ и фильтрация строк по условию Н_ОБУЧЕНИЯ.НЗК::integer < 1000, полный скан таблицы Н_ЛЮДИ и фильтрация строк по условию Н_ЛЮДИ.ИМЯ < 'Роман', соединение таблиц Н_ЛЮДИ и Н_ОБУЧЕНИЯ по условию Н_ЛЮДИ.ИД=Н_ОБУЧЕНИЯ.ЧЛВК_ИД

Происходит полный скан таблицы Н_УЧЕНИКИ и фильтрация строк по условию Н_УЧЕНИКИ.ГРУППА::integer < 4100, после этого производится соединение таблиц Н_ОБУЧЕНИЯ и Н_УЧЕНИКИ по условию Н_ОБУЧЕНИЯ.ЧЛВК_ИД=Н_УЧЕНИКИ.ЧЛВК_ИД и выбор атрибутов: Н_ЛЮДИ.ИД, Н_ОБУЧЕНИЯ.НЗК, Н_УЧЕНИКИ.ГРУППА

В данном подходе фильтрация строк происходит до соединения таблиц, получается, само соединение происходит по уже выбранным атрибутам, а не по таблицам целиком

Вывод: план №2 будет более эффективным по сравнению с планом №1

Применение индексов

```
CREATE INDEX person_id_ind ON "Н_ЛЮДИ" USING hash("ИД");  
CREATE INDEX education_person_id_ind ON "Н_ОБУЧЕНИЯ" USING hash("ЧЛВК_ИД");  
CREATE INDEX students_person_id_ind ON "Н_УЧЕНИКИ" USING hash("ЧЛВК_ИД");  
CREATE INDEX person_name_ind ON "Н_ЛЮДИ" USING btree("ИМЯ");  
CREATE INDEX education_NZK_ind ON "Н_ОБУЧЕНИЯ" USING btree("НЗК");  
CREATE INDEX students_group_ind ON "Н_УЧЕНИКИ" USING btree("ГРУППА");
```

Добавление индексов на Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_ЛЮДИ.ИД и Н_УЧЕНИКИ.ЧЛВК_ИД ускорит выполнение соединения таблиц, для этого применены hash-индексы, которые используются с оператором равенства и позволяют быстрее находить строки для объединения таблиц. Для Н_ЛЮДИ.ИМЯ, Н_ОБУЧЕНИЯ.НЗК и Н_УЧЕНИКИ.ГРУППА были выбраны b-tree индексы, которые используются для диапазонных запросов, и они подходят для фильтрации строк по условиям с WHERE

Изменения планов при добавлении индексов

Планы выполнения запросов изменятся при добавлении индексов, будет выполняться индексный скан таблицы вместо полного скана

Вывод команды EXPLAIN ANALYZE

QUERY PLAN
Hash Join (cost=382.63..1416.95 rows=2124 width=14) (actual time=7.338..10.120 rows=6 loops=1)
Hash Cond: ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
-> Seq Scan on "Н_УЧЕНИКИ" (cost=0.00..983.94 rows=7770 width=8) (actual time=0.013..4.892 rows=16365 loops=1)
Filter: (("ГРУППА")::integer < 4100)
Rows Removed by Filter: 6946
-> Hash (cost=365.14..365.14 rows=1399 width=14) (actual time=3.862..3.864 rows=2 loops=1)
Buckets: 2048 Batches: 1 Memory Usage: 17kB
-> Hash Join (cost=165.79..365.14 rows=1399 width=14) (actual time=2.632..3.861 rows=2 loops=1)
Hash Cond: ("Н_ЛЮДИ"."ИД" = "Н_ОБУЧЕНИЯ"."ЧЛВК_ИД")
-> Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=4277 width=4) (actual time=0.006..2.721 rows=4277 loops=1)
Filter: (("ИМЯ")::text < 'Роман')::text)
Rows Removed by Filter: 841
-> Hash (cost=144.87..144.87 rows=1674 width=10) (actual time=0.643..0.644 rows=2 loops=1)
Buckets: 2048 Batches: 1 Memory Usage: 17kB
-> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..144.87 rows=1674 width=10) (actual time=0.178..0.630 rows=2 loops=1)
Filter: (("НЗК")::integer < 1000)
Rows Removed by Filter: 5019
Planning Time: 0.697 ms
Execution Time: 10.193 ms
19 rows

Вывод

Во время выполнения лабораторной работы я научилась использовать индексы для повышения производительности запросов. Также изучила, как строятся планы выполнения запросов.