УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационный системы и базы данных»

Лабораторная работа №4

Вариант 777

Студент

Агнистова А.Ю.

P3125

Преподаватель

Николаев В. В.

Санкт-Петербург, 2023 г.

Цель лабораторной работы: получить теоретические и практические навыки в работе с PostgreSQL, ознакомиться с индексами и оптимизацией запросов.

Описание задания

Введите вариант: 777

Внимание! У разных вариантов разный текст задания!

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.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД.
   Фильтры (AND):
   a) Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2.
   b) Н_ВЕДОМОСТИ.ЧЛВК_ИД < 163249.
   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.ФАМИЛИЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ГРУППА.
   Фильтры: (AND)
   a) Н_ЛЮДИ.ОТЧЕСТВО > Георгиевич.
   b) Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 112514.
   c) Н_УЧЕНИКИ.ИД = 250098.
   Вид соединения: LEFT JOIN.

Запрос 1:

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД
FROM Н_ВЕДОМОСТИ
LEFT JOIN Н_ТИПЫ_ВЕДОМОСТЕЙ ON Н_ВЕДОМОСТИ.ВЕД_ИД = Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2
AND Н_ВЕДОМОСТИ.ЧЛВК_ИД < 163249;
```

Возможный индекс:

B-tree (поддерживает операции >,<).

Будет полезным добавить индекс на атрибут ЧЛВК_ИД из-за большого количества перебора при выполнении запроса. Индекс позволит сократить это время.

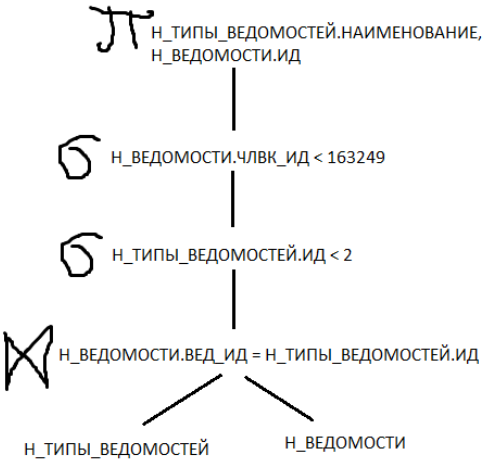
CREATE INDEX idx_type_of_statement ON Н_ВЕДОМОСТИ(ЧЛВК_ИД);

EXPLAIN ANALYZE:

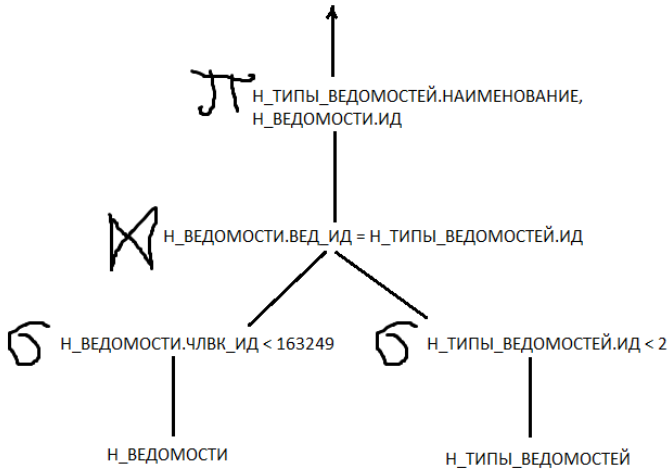
```
ucheb=> EXPLAIN ANALYZE
ucheb-> SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ИД
ucheb-> FROM Н_ВЕДОМОСТИ
ucheb-> LEFT JOIN Н_ТИПЫ_ВЕДОМОСТЕЙ ON Н_ВЕДОМОСТИ.ВЕД_ИД = Н_ВЕДОМОСТИ.ИД
ucheb-> WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД < 2
ucheb-> AND Н_ВЕДОМОСТИ.ЧЛВК_ИД < 163249;
                                                                 QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=1000.00..6579.61 rows=1112 width=422) (actual time=16.489..18.491 rows=0 loops=1)
   ->  Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ"  (cost=0.00..1.04 rows=1 width=418) (actual time=0.010..0.014 rows=1 loops=1)
         Filter: ("ИД" < 2)
         Rows Removed by Filter: 2
   ->  Gather  (cost=1000.00..6567.45 rows=1112 width=4) (actual time=16.474..18.474 rows=0 loops=1)
         Workers Planned: 2
         Workers Launched: 2
         ->  Parallel Seq Scan on "Н_ВЕДОМОСТИ"  (cost=0.00..5456.25 rows=463 width=4) (actual time=13.680..13.681 rows=0 loops=3)
               Filter: (("ЧЛВК_ИД" < 163249) AND ("ВЕД_ИД" = "ИД"))
               Rows Removed by Filter: 74147
 Planning Time: 0.203 ms
 Execution Time: 18.523 ms
(12 строк)
```

Планы выполнения запроса:

План 1:



План 2:



Более эффективным будет 1-й план запроса, так как в таком случае сначала будет проходить выборка данных, а потом объединение отношений. Также

надо учитывать, что 2 выборка небольшая из-за условия и пройдет очень быстро из-за отсеивания данных на первой выборке.

Запрос 2:

SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, 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 > 'Георгиевич'

AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД = 112514

AND Н_УЧЕНИКИ.ИД = 250098;

Возможные индексы:

B-Tree. Для индексирования строковых значений (особенно когда нужна сортировка) это самый подходящий индекс.

HASH. Поддерживает операцию =, скорость выполнения O(1), то есть константа и не зависит от объема данных.

CREATE INDEX idx_petrynomic ON Н_ЛЮДИ(ОТЧЕСТВО);

CREATE INDEX idx_human_id ON Н_ОБУЧЕНИЯ USING HASH(ЧЛВК_ИД);

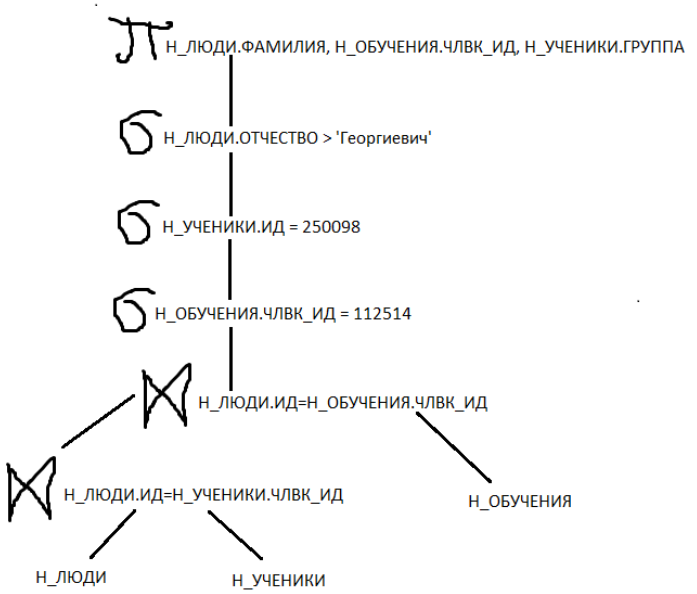
CREATE INDEX idx_student_id ON Н_УЧЕНИКИ USING HASH(ИД);

EXPLAIN ANALYZE:

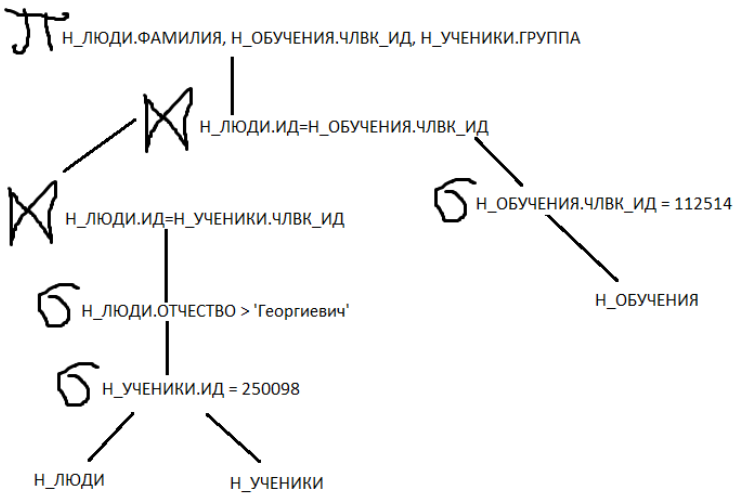
```
                                                  QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.85..20.93 rows=1 width=24) (actual time=0.024..0.024 rows=0 loops=1)
   ->  Nested Loop  (cost=0.56..12.61 rows=1 width=24) (actual time=0.023..0.024 rows=0 loops=1)
         ->  Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ"  (cost=0.28..8.30 rows=1 width=20) (actual time=0.023..0.023 rows=0 loops=1)
               Index Cond: ("ИД" = 112514)
               Filter: (("ОТЧЕСТВО")::text > 'Георгиевич'::text)
               Rows Removed by Filter: 1
         ->  Index Only Scan using "ОБУЧ_ЧЛВК_FK_I" on "Н_ОБУЧЕНИЯ"  (cost=0.28..4.30 rows=1 width=4) (never executed)
               Index Cond: ("ЧЛВК_ИД" = 112514)
               Heap Fetches: 0
   ->  Index Scan using "УЧЕН_РК" on "Н_УЧЕНИКИ"  (cost=0.29..8.31 rows=1 width=8) (never executed)
         Index Cond: ("ИД" = 250098)
         Filter: ("ЧЛВК_ИД" = 112514)
 Planning Time: 0.498 ms
 Execution Time: 0.101 ms
(14 строк)
```

Планы выполнения запроса:

План 1:



План 2:



Более эффективным будет 1-й план запроса, так как в таком случае сначала будет проходить выборка данных, а потом объединение отношений. Соединения отношение сделаны в виде левостороннего дерева.

Выводы

При выполнении лабораторной работы я познакомилась с планами выполнения запросов, индексами, командой EXPLAIN ANALYZE, оптимизировала запросы.