Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4

Вариант 8

Выполнил:

Козаченко Данил Александрович

Группа P3112

Проверил:

Кустарев Иван Павлович

Содержание

[Задание 3](#_Toc199029788)

[Запрос 1 3](#_Toc199029789)

[Реализация запроса на SQL 3](#_Toc199029790)

[Индексы 4](#_Toc199029791)

[Возможные планы выполнения запроса 4](#_Toc199029792)

[Вывод EXPLAIN ANALYZE 4](#_Toc199029793)

[Запрос 2 5](#_Toc199029794)

[Реализация запроса на SQL 5](#_Toc199029795)

[Индексы 5](#_Toc199029796)

[Возможные планы выполнения запроса 6](#_Toc199029797)

[Вывод EXPLAIN ANALYZE 6](#_Toc199029798)

[Вывод 6](#_Toc199029799)

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.ЧЛВК\_ИД.  
   Фильтры (AND):   
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Перезачет.  
   b) Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08.  
   c) Н\_ВЕДОМОСТИ.ДАТА < 2010-06-18.  
   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.ИМЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ГРУППА.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО = Сергеевич.  
   b) Н\_ОБУЧЕНИЯ.НЗК = 933232.  
   Вид соединения: INNER JOIN.

Запрос 1

Реализация запроса на SQL

//Поправил фильтр в запросе с невозможными датами

SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"  
 RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" < 'Перезачет'  
 AND "Н\_ВЕДОМОСТИ"."ДАТА" < '2022-06-08'  
 AND "Н\_ВЕДОМОСТИ"."ДАТА" > '2010-06-18';

Индексы

CREATE INDEX idx\_Н\_ТИПЫ\_ВЕДОМОСТЕЙ\_НАИМЕНОВАНИЕ ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" USING btree ("НАИМЕНОВАНИЕ");

Индекс на атрибуте НАИМЕНОВАНИЕ позволит ускорить фильтрацию по наименованию.

CREATE INDEX idx\_Н\_ВЕДОМОСТИ\_ДАТА ON "Н\_ВЕДОМОСТИ" USING btree ("ДАТА");

Индекс на атрибуте ДАТА позволит ускорить фильтрацию по дате.

CREATE INDEX idx\_Н\_ВЕДОМОСТИ\_ТВ\_ИД ON "Н\_ВЕДОМОСТИ" USING btree ("ТВ\_ИД");

Индекс на атрибуте ТВ\_ИД позволит ускорить соединение таблиц.

CREATE INDEX idx\_Н\_ТИПЫ\_ВЕДОМОСТЕЙ\_ИД ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" USING btree ("ИД");

Btree индекс на атрибуте ИД позволит ускорить соединение таблиц.

Возможные планы выполнения запроса

|  |  |
| --- | --- |
| Рисунок 1 | Рисунок 2 |

Самым оптимальным планом будет план, изображённый на Рисунок 2, так как в нём сначала производится выборка необходимых строк по условиям и только потом происходит соединение таблиц с помощью RIGHT JOIN.

При добавлении индексов план выполнения запросов изменится, так как будет выполняться индексный скан. Также будет использован более быстрый Hash Join, благодаря индексам на ТВ\_ИД.

Вывод EXPLAIN ANALYZE

Hash Join (cost=362.81..5019.32 rows=8633 width=8) (actual time=0.878..10.612 rows=18967 loops=1)

Hash Cond: ("Н\_ВЕДОМОСТИ"."ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")

-> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=361.76..4854.24 rows=25899 width=8) (actual time=0.823..4.815 rows=25728 loops=1)

Recheck Cond: (("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone) AND ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone))

Heap Blocks: exact=651

-> Bitmap Index Scan on "ВЕД\_ДАТА\_I" (cost=0.00..355.29 rows=25899 width=0) (actual time=0.746..0.746 rows=25728 loops=1)

Index Cond: (("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone) AND ("ДАТА" > '2010-06-18 00:00:00'::timestamp without time zone))

-> Hash (cost=1.04..1.04 rows=1 width=4) (actual time=0.045..0.046 rows=1 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

-> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.017..0.020 rows=1 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text < 'Перезачет'::text)

Rows Removed by Filter: 2

Planning Time: 0.266 ms

Execution Time: 11.520 ms

Запрос 2

Реализация запроса на SQL

SELECT "Н\_ЛЮДИ"."ИМЯ", "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД", "Н\_УЧЕНИКИ"."ГРУППА"  
FROM "Н\_ЛЮДИ"  
 JOIN "Н\_ОБУЧЕНИЯ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД"  
 JOIN "Н\_УЧЕНИКИ" ON "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД"  
WHERE "Н\_ЛЮДИ"."ОТЧЕСТВО" = 'Сергеевич'  
 AND "Н\_ОБУЧЕНИЯ"."НЗК" = '933232';

Индексы

CREATE INDEX idx\_Н\_ЛЮДИ\_ОТЧЕСТВО ON "Н\_ЛЮДИ" USING btree ("ОТЧЕСТВО");

Btree индекс на атрибуте ОТЧЕСТВО позволит ускорить фильтрацию.

CREATE INDEX idx\_Н\_ОБУЧЕНИЯ\_НЗК ON "Н\_ОБУЧЕНИЯ" USING btree ("НЗК");

Btree индекс на атрибуте НЗК позволит ускорить фильтрацию.

CREATE INDEX idx\_Н\_ОБУЧЕНИЯ\_ЧЛВК\_ИД ON "Н\_ОБУЧЕНИЯ" USING btree ("ЧЛВК\_ИД");

Btree индекс на атрибуте ЧЛВК\_ИД позволит ускорить соединение таблиц.

CREATE INDEX idx\_Н\_УЧЕНИКИ\_ЧЛВК\_ИД ON "Н\_УЧЕНИКИ" USING btree ("ЧЛВК\_ИД");

Btree индекс на атрибуте ЧЛВК\_ИД позволит ускорить соединение таблиц.

CREATE INDEX idx\_Н\_ЛЮДИ\_ИД ON "Н\_ЛЮДИ" USING btree ("ИД");

Btree индекс на атрибуте ИД позволит ускорить соединение таблиц.

Возможные планы выполнения запроса

|  |  |
| --- | --- |
| Рисунок 1 | Рисунок 2 |

Самым оптимальным планом будет план, изображённый на Рисунок 2, так как в нём сначала производится выборка необходимых строк по условиям и только потом происходит соединение таблиц с помощью INNER JOIN.

При добавлении индексов план выполнения запросов изменится, так как будет выполняться индексный скан. Также Nested Loop Join станет быстрее благодаря индексам на ЧЛВК\_ИД.

Вывод EXPLAIN ANALYZE

Nested Loop (cost=0.57..134.71 rows=5 width=21) (actual time=0.049..0.558 rows=3 loops=1)

-> Nested Loop (cost=0.28..128.11 rows=1 width=21) (actual time=0.023..0.511 rows=1 loops=1)

-> Seq Scan on "Н\_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=1 width=4) (actual time=0.012..0.499 rows=1 loops=1)

Filter: (("НЗК")::text = '933232'::text)

Rows Removed by Filter: 5020

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=17) (actual time=0.008..0.008 rows=1 loops=1)

Index Cond: ("ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")

Filter: (("ОТЧЕСТВО")::text = 'Сергеевич'::text)

-> Index Scan using "УЧЕН\_ОБУЧ\_FK\_I" on "Н\_УЧЕНИКИ" (cost=0.29..6.55 rows=5 width=8) (actual time=0.024..0.044 rows=3 loops=1)

Index Cond: ("ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")

Planning Time: 1.373 ms

Execution Time: 0.596 ms

Вывод

Я подробно познакомился с базами данных и оптимизацией запросов.