Санкт-Петербургский национально исследовательский университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники



**Лабораторная работа №4 по предмету**

**“Информационные системы и базы данных”**

Вариант №313093

Выполнил: Балтабаев Дамир

Группа: P33121

Преподаватель: Шешуков Дмитрий Михайлович

г. Санкт-Петербург

2022

**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).

**Выполнение**

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.ИД < 3.  
   b) Н\_ВЕДОМОСТИ.ИД < 1250972.  
   Вид соединения: RIGHT JOIN.

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"."ИД" < 3)  
 AND ("Н\_ВЕДОМОСТИ"."ИД" < 1250972);

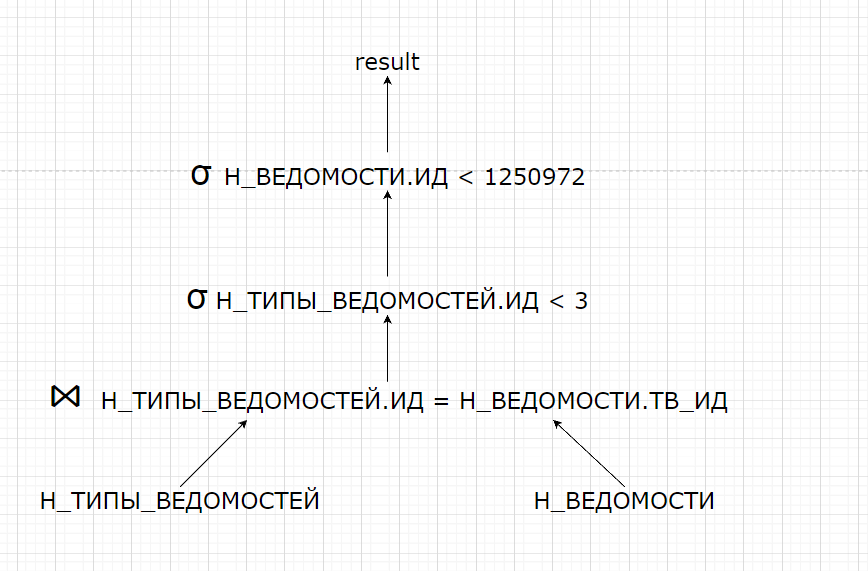
Для данного запроса идеально подходят btree индексы на таблицы Н\_ВЕДОМОСТИ и Н\_ТИПЫ\_ВЕДОМОСТЕЙ по полям Н\_ВЕДОМОСТИ.ИД и Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, т.к данные атрибуты выбранных таблиц используются в операторах WHERE и JOIN, где они сравниваются путем использования логических операций (в моем случае “<”). Но т.к выбранные мною атрибуты являются первичными ключами, то нет смысла создавать btree индексы на них, ибо таблицы и так имеют их по умолчанию. CREATE INDEX ON “Н\_ВЕДОМОСТИ” USING btree(“ИД”); CREATE INDEX ON “Н\_ТИПЫ\_ВЕДОМОСТЕЙ” USING btree(“ИД”);

Также имеет смысл создать hash индекс таблице Н\_ВЕДОМОСТИ по атрибуту ТВ\_ИД, чтобы ускорить выборку данных в RIGHT JOIN;

CREATE INDEX ON “Н\_ВЕДОМОСТИ” USING btree(“ТВ\_ИД”);

Изображение выглядит как стол

Автоматически созданное описание



Первый порядок выполнения является оптимальным, т.к изначально происходит выбор нужных записей, попадающих под определенные условия, а уже после этого происходит объединение. Размер промежуточных данных меньше, значит план является оптимальным.

EXPLAIN ANALYZE 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"."ИД" < 3)  
 AND ("Н\_ВЕДОМОСТИ"."ИД" < 1250972);

Nested Loop (cost=828.76..6662.71 rows=65470 width=8) (actual time=7.394..84.807 rows=189498 loops=1)

" -> Seq Scan on ""Н\_ТИПЫ\_ВЕДОМОСТЕЙ"" (cost=0.00..1.04 rows=1 width=4) (actual time=0.018..0.021 rows=2 loops=1)"

" Filter: (""ИД"" < 3)"

Rows Removed by Filter: 1

" -> Bitmap Heap Scan on ""Н\_ВЕДОМОСТИ"" (cost=828.76..6006.97 rows=65470 width=8) (actual time=4.298..29.756 rows=94749 loops=2)"

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

" Filter: (""ИД"" < 1250972)"

Rows Removed by Filter: 11434

Heap Blocks: exact=6314

" -> Bitmap Index Scan on ""ВЕД\_ТВ\_FK\_I"" (cost=0.00..812.40 rows=74147 width=0) (actual time=3.807..3.807 rows=106182 loops=2)"

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

Planning Time: 1.280 ms

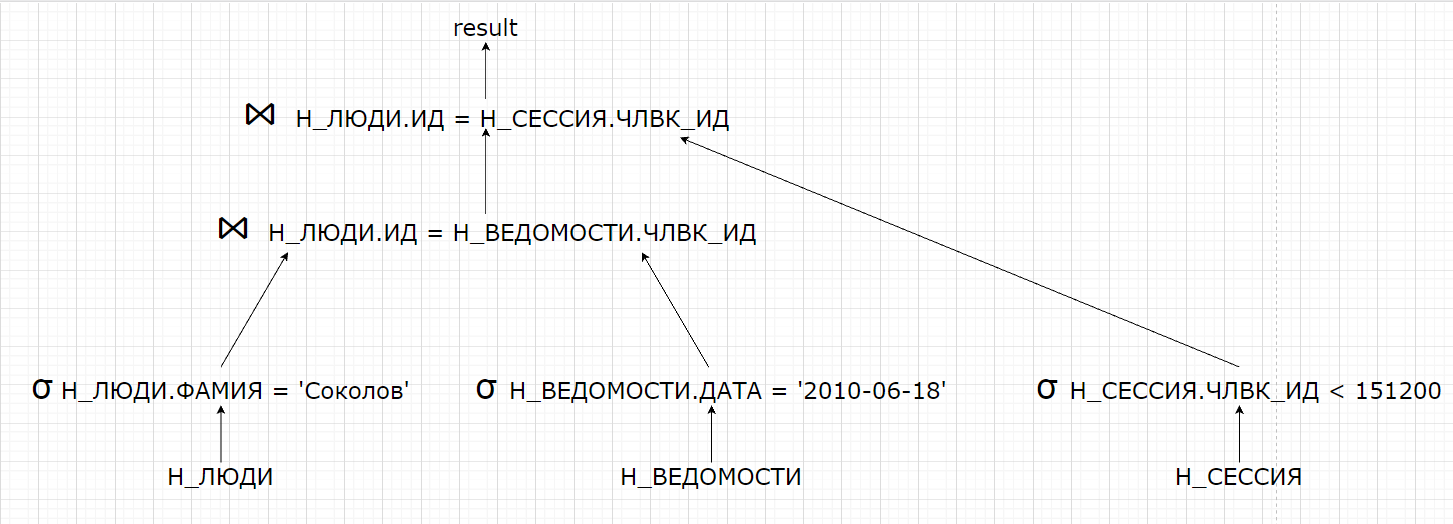
Execution Time: 93.670 ms

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.УЧГОД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ФАМИЛИЯ = Соколов.  
   b) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.  
   c) Н\_СЕССИЯ.ЧЛВК\_ИД < 151200.  
   Вид соединения: INNER JOIN.

SELECT "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД"  
FROM "Н\_ЛЮДИ"  
 JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
 JOIN "Н\_СЕССИЯ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE ("Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Соколов')  
 AND ("Н\_ВЕДОМОСТИ"."ДАТА" = '2010-06-18')  
 AND ("Н\_СЕССИЯ"."ЧЛВК\_ИД" < 151200);

Для данного запроса идеально подходят следующие индексы:

* hash индекс для таблицы Н\_ЛЮДИ по атрибуту ФАМИЛИЯ, т.к это позволит быстрее отбирать необходимые записи и сравнивать их операцией ‘=’, поэтому hash индекс подходит лучше всего. CREATE INDEX ON “Н\_ЛЮДИ” USING hash(“ФАМИЛИЯ”).
* hash индекс для таблицы Н\_СЕССИЯ по атрибуту ДАТА, т.к это позволит быстрее отбирать необходимые записи и сравнивать их операцией ‘=’, поэтому hash индекс подходит лучше всего. CREATE INDEX ON “Н\_СЕССИЯ” USING hash(“ДАТА”).
* btree индекс для таблицы Н\_СЕССИЯ по атрибуту ЧЛВК\_ИД, т.к это позволит быстрее отбирать необходимые записи и сравнивать их операциями ‘=, <’, поэтому btree индекс подходит лучше всего. CREATE INDEX ON “Н\_СЕССИЯ” USING btree(“ЧЛВК\_ИД”).



Изображение выглядит как стол

Автоматически созданное описание

Первый порядок выполнения является оптимальным, т.к изначально происходит выбор нужных записей, попадающих под определенные условия, а уже после этого происходит объединение. Размер промежуточных данных меньше, значит план является оптимальным.

EXPLAIN ANALYZE SELECT "Н\_ЛЮДИ"."ФАМИЛИЯ", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД"  
FROM "Н\_ЛЮДИ"  
 JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
 JOIN "Н\_СЕССИЯ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE ("Н\_ЛЮДИ"."ФАМИЛИЯ" = 'Соколов')  
 AND ("Н\_ВЕДОМОСТИ"."ДАТА" = '2010-06-18')  
 AND ("Н\_СЕССИЯ"."ЧЛВК\_ИД" < 151200)

Nested Loop (cost=45.32..225.71 rows=1 width=34) (actual time=1.000..1.002 rows=0 loops=1)

" Join Filter: (""Н\_ЛЮДИ"".""ИД"" = ""Н\_ВЕДОМОСТИ"".""ЧЛВК\_ИД"")"

-> Hash Join (cost=45.03..171.43 rows=8 width=34) (actual time=0.999..1.001 rows=0 loops=1)

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

" -> Seq Scan on ""Н\_СЕССИЯ"" (cost=0.00..117.90 rows=3237 width=14) (actual time=0.010..0.654 rows=3237 loops=1)"

" Filter: (""ЧЛВК\_ИД"" < 151200)"

Rows Removed by Filter: 515

-> Hash (cost=44.85..44.85 rows=14 width=20) (actual time=0.068..0.069 rows=14 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 9kB

" -> Bitmap Heap Scan on ""Н\_ЛЮДИ"" (cost=4.39..44.85 rows=14 width=20) (actual time=0.033..0.056 rows=14 loops=1)"

" Recheck Cond: ((""ФАМИЛИЯ"")::text = 'Соколов'::text)"

Heap Blocks: exact=13

" -> Bitmap Index Scan on ""ФАМ\_ЛЮД"" (cost=0.00..4.39 rows=14 width=0) (actual time=0.026..0.026 rows=14 loops=1)"

" Index Cond: ((""ФАМИЛИЯ"")::text = 'Соколов'::text)"

" -> Index Scan using ""ВЕД\_ЧЛВК\_FK\_IFK"" on ""Н\_ВЕДОМОСТИ"" (cost=0.29..6.77 rows=1 width=12) (never executed)"

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

" Filter: (""ДАТА"" = '2010-06-18 00:00:00'::timestamp without time zone)"

Planning Time: 0.612 ms

Execution Time: 1.061 ms

При добавлении индексации планы выполнения изменятся, т.к вместо обычного SEQ Scan могут использоваться Index Scan, Index only Scan или же Bitmap index Scan;

**Вывод**

В ходе выполнения данной лабораторной работы я познакомился с понятием индексов, их разновидностями и тем, как они влияют на быстродействие. Также описал возможные планы выполнения запросов.