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 1466*

Студент

Армут Алина

*P3114*

Преподаватель:

Горбунов Михаил

Санкт-Петербург, 2023 г.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

1)

SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"  
FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" "Н\_ВЕДОМОСТИ"  
RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ВЕДОМОСТИ"."ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"  
WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" < 'Экзаменационный лист'  
AND "Н\_ВЕДОМОСТИ"."ДАТА" < '1998-01-05'  
AND "Н\_ВЕДОМОСТИ"."ДАТА" = '2022-06-08';

HASH. Поддерживает операцию “=” ( но не поддерживают операции сравнения <,>) скорость выполнения О(1), то есть константа и не зависит от объема данных.

Поэтому в данным случае используем тип индекса B-tree.

CREATE INDEX НАИМЕНОВАНИЕ\_ИНДЕКС   
ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ("НАИМЕНОВАНИЕ");

Будет эффективно использовать этот индекс так как: из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор “<” и данный атрибут будет редко изменяться, а значит, индекс не придется часто менять. Индекс позволит сократить время выполнения запроса.

CREATE INDEX index\_date  
ON "Н\_ВЕДОМОСТИ" USING HASH ("ДАТА");

Будет полезным добавить индекс на атрибут "ДАТА" из-за большого количества перебора(то есть 500+) при выполнении запроса. Индекс позволит сократить время выполнения запроса.

Также из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор “=”.

Вывод команды EXPLAIN ANALYZE:

Nested Loop (cost=0.29..8.25 rows=1 width=422) (actual time=0.003..0.004 rows=0 loops=1)

Join Filter: ("Н\_ВЕДОМОСТИ"."КОГДА\_СОЗДАЛ" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."КОГДА\_СОЗДАЛ")

-> Index Scan using "ВЕД\_ДАТА\_I" on "Н\_ВЕДОМОСТИ" (cost=0.29..7.20 rows=1 width=12) (actual time=0.002..0.002 rows=0 loops=1)

Index Cond: (("ДАТА" < '1998-01-05 00:00:00'::timestamp without time zone) AND ("ДАТА" = '2022-06-08 00:00:00'::timestamp without time zone))

-> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=426) (never executed)

Filter: (("НАИМЕНОВАНИЕ")::text < 'Экзаменационный лист'::text)

Planning Time: 1.140 ms

Execution Time: 0.061 ms

(8 строк)

План выполнения запроса:

ПЛАН 1

RESULT



SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ"< Экзаменационный экзамен



"Н\_ВЕДОМОСТИ"."ДАТА"< 1998-01-05



"Н\_ВЕДОМОСТИ"."ДАТА" = 2022-06-08



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"

"Н\_ВЕДОМОСТИ"

ПЛАН 2

RESULT



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"



"Н\_ВЕДОМОСТИ"."ДАТА" = 2022-06-08



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ"< Экзаменационный экзамен



"Н\_ВЕДОМОСТИ"."ДАТА"< 1998-01-05



"Н\_ВЕДОМОСТИ"

"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"

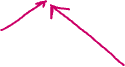


RESULT

SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"



RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД"



"Н\_ВЕДОМОСТИ"."ДАТА" = 2022-06-08



"Н\_ВЕДОМОСТИ"."ДАТА"< 1998-01-05

"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ"< Экзаменационный экзамен



SELECT "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_ВЕДОМОСТИ"."ТВ\_ИД"

SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД"



"Н\_ТИПЫ\_ВЕДОМОСТЕЙ"

"Н\_ВЕДОМОСТИ"

Более эффективным будет данный план запроса, так как в таком случае проекция сделана раньше соединения, выборка сделана раньше соединения, минимизирован размер промежуточных данных.

Соединения отношение сделаны в виде правостороннего дерева.

2)

SELECT "Н\_ЛЮДИ"."ИД", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД"  
FROM "Н\_ЛЮДИ"  
INNER JOIN "Н\_ВЕДОМОСТИ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"

INNER JOIN "Н\_СЕССИЯ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"  
WHERE "Н\_ЛЮДИ"."ИД" < 100865  
AND "Н\_ВЕДОМОСТИ"."ДАТА" > '2010-06-18';

CREATE INDEX index\_people ON "Н\_ЛЮДИ"("ИД");

Будет эффективно использовать этот индекс так как: из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор “<” и данный атрибут будет редко изменяться, а значит, индекс не придется часто менять.

CREATE INDEX index\_date ON "Н\_ВЕДОМОСТИ" ("ДАТА");

Будет полезным добавить индекс на атрибут "ДАТА" из-за большого количества перебора(то есть 500+) при выполнении запроса. Индекс позволит сократить время выполнения запроса.

Также из-за того, что данный атрибут используется в фильтрации вместе с WHERE, запрос использует оператор “>”.

Вывод команды EXPLAIN ANALYZE:

Nested Loop (cost=5.36..223.16 rows=104 width=22) (actual time=3.174..3.176 rows=0 loops=1)

-> Hash Join (cost=5.06..123.44 rows=16 width=18) (actual time=0.057..1.366 rows=1641 loops=1)

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

-> Seq Scan on "Н\_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=14) (actual time=0.008..0.425 rows=3752 loops=1)

-> Hash (cost=4.74..4.74 rows=26 width=4) (actual time=0.022..0.023 rows=40 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 10kB

-> Index Only Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..4.74 rows=26 width=4) (actual time=0.004..0.009 rows=40 loops=1)

Index Cond: ("ИД" < 100865)

Heap Fetches: 0

-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..6.15 rows=8 width=12) (actual time=0.001..0.001 rows=0 loops=1641)

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

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

Planning Time: 1.944 ms

План выполнения запроса:

ПЛАН 1

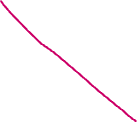
RESULT



SELECT "Н\_ЛЮДИ"."ИД", "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_СЕССИЯ"."УЧГОД"



INNER JOIN "Н\_СЕССИЯ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД"



INNER JOIN "Н\_ВЕДОМОСТИ" ON "Н\_СЕССИЯ"."ЧЛВК\_ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"



"Н\_ВЕДОМОСТИ"."ДАТА" > '2010-06-18'



"Н\_ЛЮДИ"."ИД" < 100865

"Н\_ЛЮДИ"."ИД"



SELECT "Н\_СЕССИЯ"."УЧГОД","Н\_СЕССИЯ"."ЧЛВК\_ИД"

SELECT "Н\_ВЕДОМОСТИ"."ДАТА", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"



SELECT "Н\_ЛЮДИ"."ИД"



"Н\_ЛЮДИ"

"Н\_СЕССИЯ"

"Н\_ВЕДОМОСТИ"

Данный план является оптимальным, так как:

1. Проекция сделана раньше соединения;
2. Выборка сделана раньше соединения;
3. Минимизирован размер промежуточных данных.

Выводы

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