

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационные системы и базы данных»

Лабораторная работа №4

Вариант 40273

Студент

Левченко Ярослав

Р3118

Преподаватель

Инячина Диана

Санкт-Петербург, 2023 г.

Описание 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 состояются 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.ИМЯ, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ЛЮДИ.ИД > 152862.
б) Н_ВЕДОМОСТИ.ДАТА < 1998-01-05.
Вид соединения: LEFT JOIN.
2. 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.ЧЛВК_ИД, Н_СЕССИЯ.ДАТА.
Фильтры (AND):
а) Н_ЛЮДИ.ИД > 100012.
б) Н_ВЕДОМОСТИ.ИД < 1250981.

с) Н_СЕССИЯ.ДАТА < 2002-01-04.

Вид соединения: INNER JOIN.

Реализация первого запроса

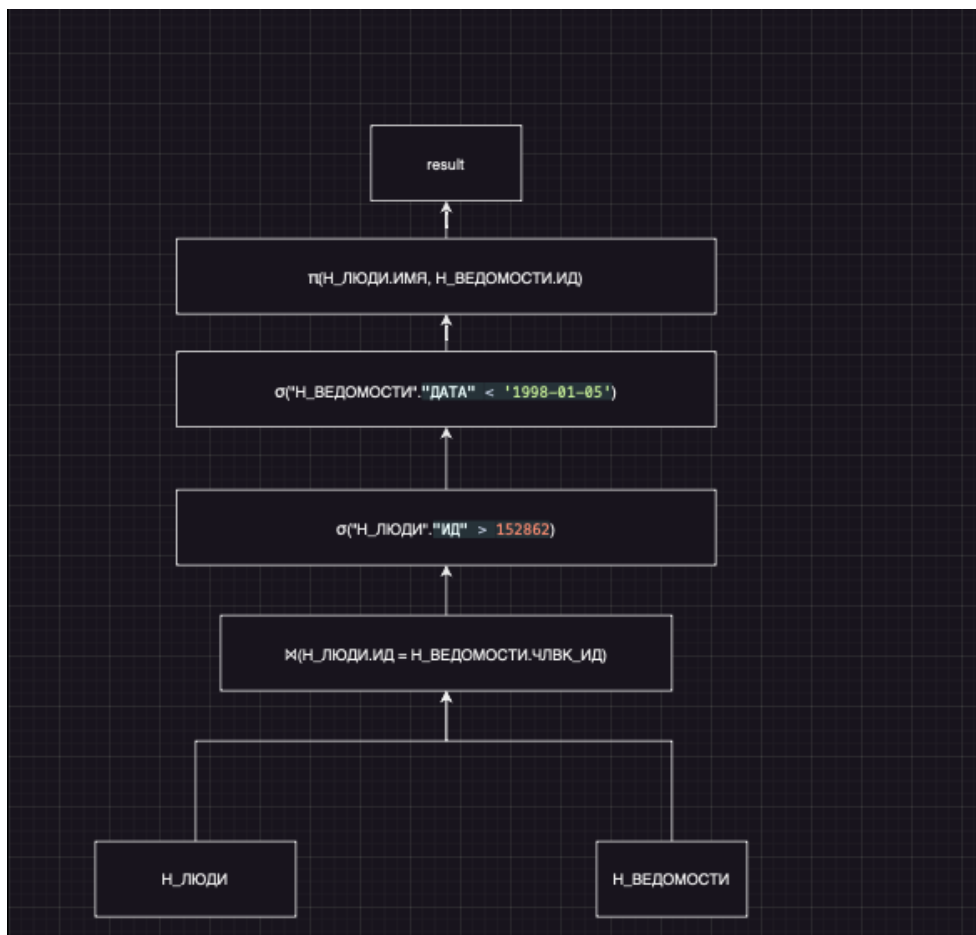
1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:
Таблицы: Н_ЛЮДИ, Н_ВЕДОМОСТИ.
Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_ВЕДОМОСТИ.ИД.
Фильтры (AND):
а) Н_ЛЮДИ.ИД > 152862.
б) Н_ВЕДОМОСТИ.ДАТА < 1998-01-05.
Вид соединения: LEFT JOIN.

Запрос:

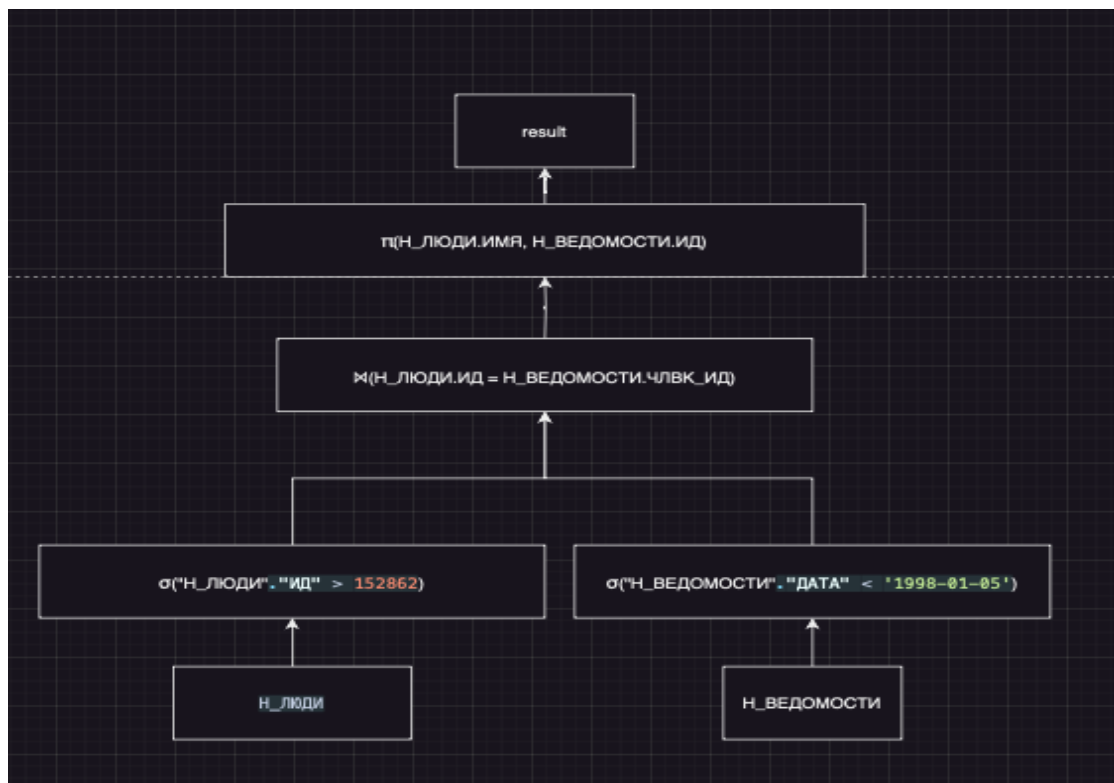
```
8  
9 ✓ SELECT "Н_люди"."ИМЯ", "Н_ВЕДОМОСТИ"."ИД"  
10 FROM "Н_люди"  
11 LEFT JOIN "Н_ВЕДОМОСТИ" on "Н_ВЕДОМОСТИ"."ЧЛВК_ИД" = "Н_люди"."ИД"  
12 WHERE "Н_люди"."ИД" > 152862  
13 AND "Н_ВЕДОМОСТИ"."ДАТА" < '1998-01-05';
```

Планы выполнения:

а)



б)



в)



Оптимальный план - б, т.к. сначала производится выбор строк по указанным условиям, а только потом отфильтрованные строки соединяются с помощью left join.

Индексы:

```

-- предикат "Н_ВЕДОМОСТИ"."ДАТА" < '1998-01-05' -> B-tree
CREATE INDEX "ИНД_ВЕД_ДАТА" ON "Н_ВЕДОМОСТИ" USING btree ("ДАТА");
-- предикат "Н_ЛЮДИ"."ИД" > 152862 < '1998-01-05' -> B-tree
CREATE INDEX "ИНД_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING btree ("ИД");

```

Вывод Explain Analyze:

	QUERY PLAN
1	Nested Loop (cost=0.58..15.52 rows=1 width=17) (actual time=0.004..0.005 rows=0 loops=1)
2	-> Index Scan using "ВЕД_ДАТА_I" on "Н_ВЕДОМОСТИ" (cost=0.29..7.20 rows=1 width=8) (actual time=0.004..0.004 rows=0 loops=1)
3	Index Cond: ("ДАТА" < '1998-01-05 00:00:00'::timestamp without time zone)
4	-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=17) (never executed)
5	Index Cond: (("ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД") AND ("ИД" > 152862))
6	Planning Time: 0.291 ms
7	Execution Time: 0.032 ms

Реализация второго запроса

Запрос:

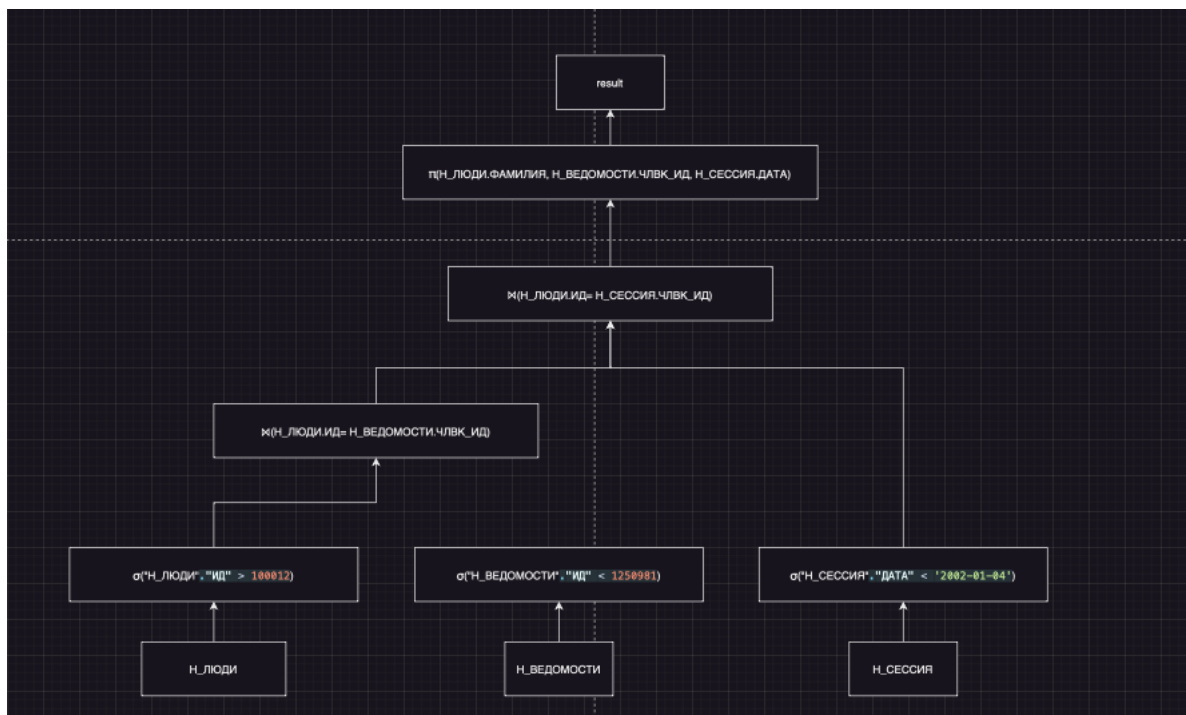
```
-- Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
-- Таблицы: Н_люди, Н_ВЕДОМОСТИ, Н_СЕССИЯ.  
-- Вывести атрибуты: Н_люди.ФАМИЛИЯ, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.ДАТА.  
-- Фильтры (AND):  
-- а) Н_люди.ИД > 100012.  
-- б) Н_ВЕДОМОСТИ.ИД < 1250981.  
-- с) Н_СЕССИЯ.ДАТА < 2002-01-04.  
-- Вид соединения: INNER JOIN.
```

```
EXPLAIN ANALYZE SELECT "Н_люди"."ФАМИЛИЯ", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД", "Н_СЕССИЯ"."ДАТА"  
FROM "Н_люди"  
    INNER JOIN "Н_ВЕДОМОСТИ" ON "Н_люди"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"  
    INNER JOIN "Н_СЕССИЯ" ON "Н_люди"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"  
WHERE "Н_люди"."ИД" > 100012  
AND "Н_ВЕДОМОСТИ"."ИД" < 1250981  
AND "Н_СЕССИЯ"."ДАТА" < '2002-01-04'
```

a)



б)



Оптимальным является план б, потому что все три таблицы имеют большое количество строк, а после выполнения последнего соединения их останется небольшое количество. За счет использования проекции и фильтрации на ранних этапах происходит соединение только нужных нам атрибутов, следовательно промежуточные данные меньше.

Индексы:

```

-- предикат "H_ЛЮДИ"."ИД" > 100012 -> B-tree
CREATE INDEX "ИНД_ЧЕЛ_ИД" ON "H_ЛЮДИ" USING btree ("ИД");
-- предикат "H_ВЕДОМОСТИ"."ИД" < 1250981 -> B-tree
CREATE INDEX "ИНД_ВЕД_ИД" ON "H_ВЕДОМОСТИ" USING btree ("ИД");
-- предикат "H_СЕССИЯ"."ДАТА" < '2002-01-04' -> B-tree
CREATE INDEX "ИНД_ДАТА" ON "H_СЕССИЯ" USING btree ("ДАТА");

```

Вывод Explain Analyze:

	QUERY PLAN
1	Nested Loop (cost=0.58..131.79 rows=33 width=28) (actual time=0.449..0.450 rows=0 loops=1)
2	-> Nested Loop (cost=0.28..126.21 rows=1 width=32) (actual time=0.449..0.450 rows=0 loops=1)
3	-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=1 width=12) (actual time=0.449..0.449 rows=0 loops=1)
4	Filter: ("ДАТА" < '2002-01-04 00:00:00'::timestamp without time zone)
5	Rows Removed by Filter: 3752
6	-> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=20) (never executed)
7	Index Cond: (("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД") AND ("ИД" > 100012))
8	-> Index Scan using "ВЕД_ЧЛВК_ФК_ИФК" on "Н_ВЕДОМОСТИ" (cost=0.29..4.99 rows=60 width=4) (never executed)
9	Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")
10	Filter: ("ИД" < 1250981)
11	Planning Time: 0.672 ms
12	Execution Time: 0.493 ms

Выводы

При выполнении лабораторной работы я познакомилась с особенностями составления и обработки планов СУБД при использовании и без использования индексов. Были изучены основные виды индексов и стратегии соединения таблиц, применяемых в данной СУБД.