

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский университет ИТМО»
Факультет Программной инженерии и компьютерной техники

Лабораторная работа №4
ИСиБД

Вариант №11203

Группа: P33312

Выполнил: Цю Тяньшэн

Проверил(-а):
Наумова Н.А.

Санкт-Петербург
2023г

Содержание

Текст задания.....	3
Выполнение задания.....	5
Запрос №1	5
Запрос №2	6
Вывод.....	9

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.

Изменяться 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_ОЦЕНКИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ОЦЕНКИ.ПРИМЕЧАНИЕ,
Н_ВЕДОМОСТИ.ЧЛВК_ИД.

Фильтры (AND):

а) Н_ОЦЕНКИ.ПРИМЕЧАНИЕ = неудовлетворительно.

б) Н_ВЕДОМОСТИ.ИД > 1250981.

Вид соединения: INNER 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.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ЧЛВК_ИД,
Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

- a) Н_ЛЮДИ.ИМЯ > Владимир.
- b) Н_ВЕДОМОСТИ.ДАТА = 2010-06-18.
- c) Н_СЕССИЯ.ДАТА = 2012-01-25.

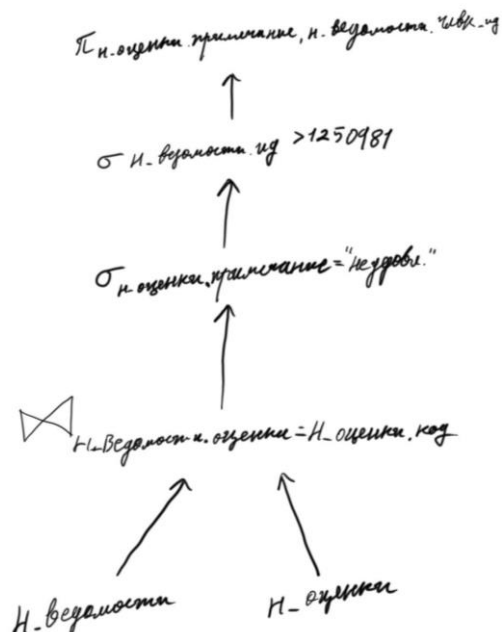
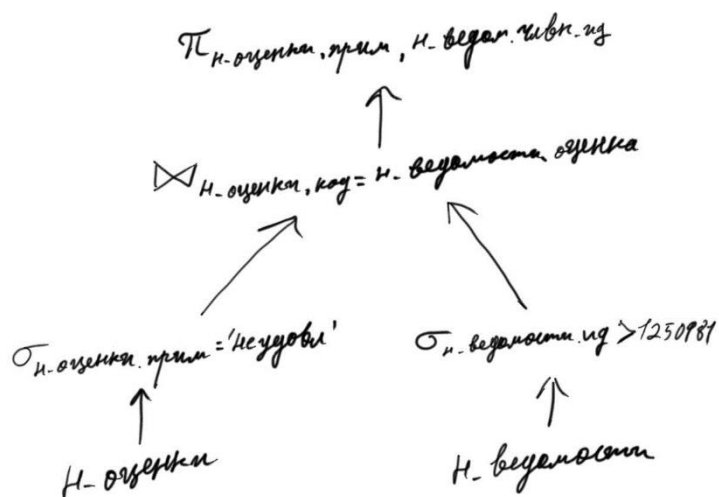
Вид соединения: RIGHT JOIN.

Выполнение задания

Запрос №1

```
SELECT Н_ОЦЕНКИ.ПРИМЕЧАНИЕ, Н_ВЕДОМОСТИ.ЧЛВК_ИД
FROM Н_ОЦЕНКИ
JOIN Н_ВЕДОМОСТИ
ON Н_ВЕДОМОСТИ.ОЦЕНКА = Н_ОЦЕНКИ.КОД
WHERE Н_ОЦЕНКИ.ПРИМЕЧАНИЕ = 'неудовлетворительно'
AND Н_ВЕДОМОСТИ.ИД > 1250981;
```

Планы выполнения



Во втором плане происходит объединение только нужной выборки, а не всех сущностей. Размер промежуточных данных меньше, значит этот план является оптимальным.

Анализ запроса

```
----- QUERY PLAN -----
Nested Loop (cost=765.57..4740.94 rows=2902 width=26)
-> Seq Scan on "Н_ОЦЕНКИ" (cost=0.00..1.11 rows=1 width=27)
    Filter: (("ПРИМЕЧАНИЕ")::text = 'неудовлетворительно'::text)
-> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=765.57..4710.81 rows=2902 width=10)
    Recheck Cond: (((("ОЦЕНКА")::text = ("Н_ОЦЕНКИ"."КОД")::text) AND ("ИД" > 1250981)))
-> BitmapAnd (cost=765.57..765.57 rows=2902 width=0)
    -> Bitmap Index Scan on "ВЕД_ОЦЕНКА_I" (cost=0.00..273.79 rows=24716 width=0)
        Index Cond: (("ОЦЕНКА")::text = ("Н_ОЦЕНКИ"."КОД")::text)
    -> Bitmap Index Scan on "ВЕД_PK" (cost=0.00..484.28 rows=26114 width=0)
        Index Cond: ("ИД" > 1250981)
```

Индексы

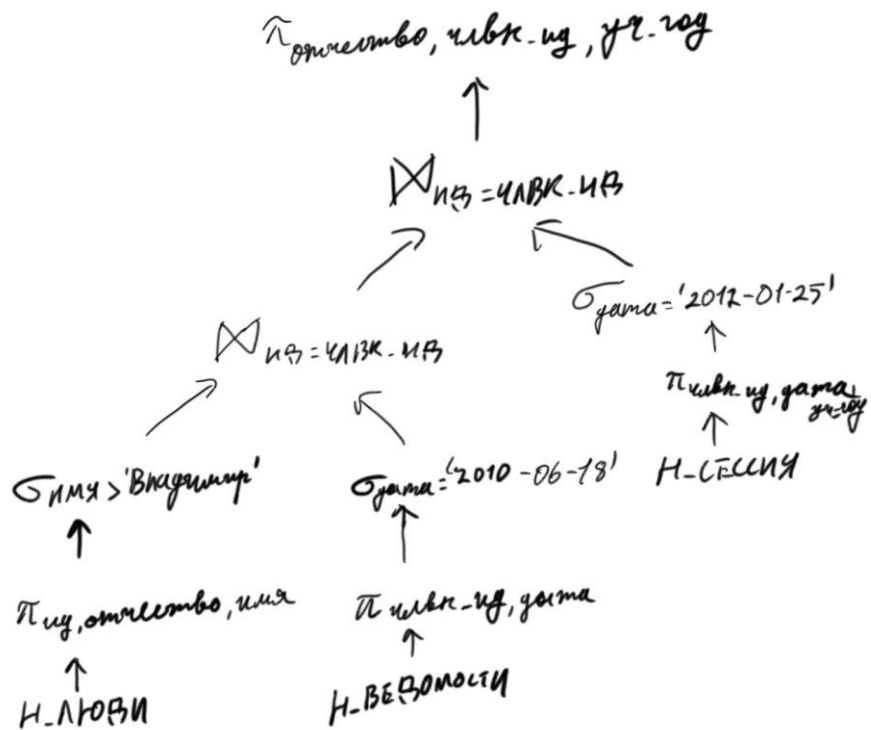
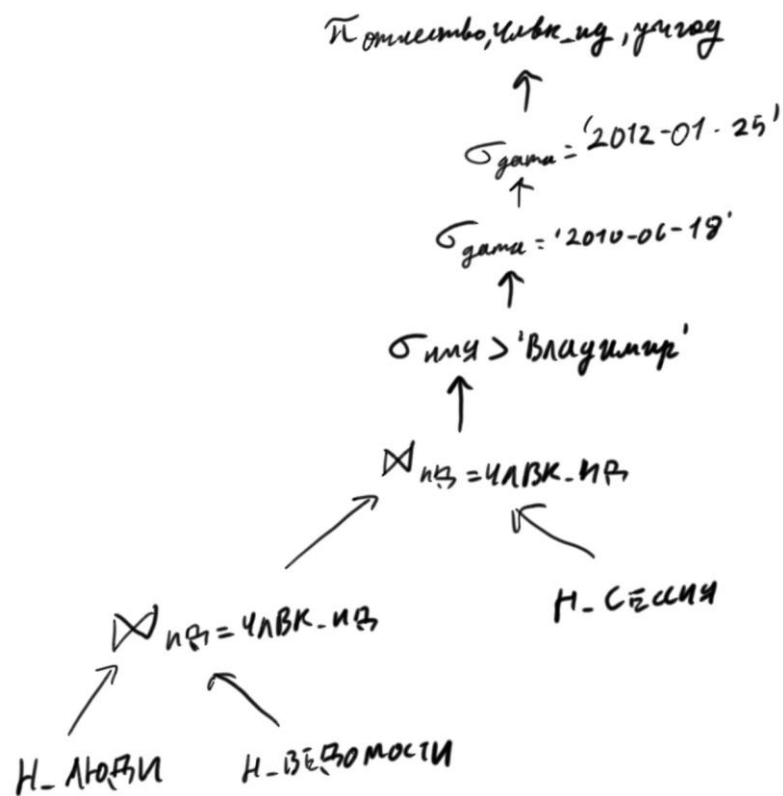
```
CREATE INDEX н_ведомости_оценка ON Н_ВЕДОМОСТИ USING
HASH(ОЦЕНКА);
CREATE INDEX н_ведомости_ид ON Н_ВЕДОМОСТИ USING btree(ИД);
```

Соединение таблиц по оценке – оптимальнее использовать HASH, а для получение нужных ведомостей с помощью оператора по ид – оптимальнее использовать btree.

Запрос №2

```
SELECT Н_люди.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.УЧГОД
FROM Н_люди
RIGHT JOIN Н_ВЕДОМОСТИ ON Н_люди.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
RIGHT JOIN Н_СЕССИЯ ON Н_люди.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE Н_люди.ИМЯ > 'Владимир'
AND Н_ВЕДОМОСТИ.ДАТА = '2010-06-18'
AND Н_СЕССИЯ.ДАТА = '2012-01-25';
```

Планы выполнения



Второй план является оптимальным, поскольку выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

Индексы

```
CREATE INDEX н_люди_имя ON Н_ЛЮДИ USING btree (ИМЯ);  
CREATE INDEX н_ведомости_дата ON Н_ВЕДОМОСТИ USING btree  
(ДАТА);  
CREATE INDEX н_сессия_дата ON Н_СЕССИЯ USING btree (ДАТА);  
CREATE INDEX н_люди_ид ON Н_ЛЮДИ USING HASH (ИД);
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

Анализ запроса

```
CREATE INDEX н_люди_имя ON Н_ЛЮДИ USING btree (ИМЯ);  
CREATE INDEX н_ведомости_дата ON Н_ВЕДОМОСТИ USING btree (ДАТА);  
CREATE INDEX н_сессия_дата ON Н_СЕССИЯ USING btree (ДАТА);  
CREATE INDEX н_люди_ид ON Н_ЛЮДИ USING HASH (ИД);  
-----  
Nested Loop (cost=0.58..179.74 rows=1 width=34)  
-> Nested Loop (cost=0.28..159.42 rows=3 width=38)  
-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..117.90 rows=5 width=14)  
Filter: ("ДАТА" = '2012-01-25 00:00:00'::timestamp without time zone)  
-> Index Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24)  
Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")  
Filter: (("ИМЯ")::text > 'Владимир'::text)  
-> Index Scan using "ВЕД_ЧЛВК_ФК_ИФК" on "Н_ВЕДОМОСТИ" (cost=0.29..6.76 rows=1 width=4)  
Index Cond: ("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")  
Filter: ("ДАТА" = '2010-06-18 00:00:00'::timestamp without time zone)
```


Вывод

При выполнении лабораторной работы я познакомился с использованием индексов для ускорения обработки запросов в SQL. Изучил, как поэтапно происходит выполнение запроса, какие планы могут использоваться и как индексы влияют на процесс.