

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4

Выполнил:
Решетников Сергей Евгеньевич
Группа Р3108
Проверила:
Преподаватель ФПиИКТ
Заболотняя Ольга Михайловна

Санкт-Петербург 2025

Оглавление

1. Задание.....	3
2. Запрос 1.....	3
2.1 Планы выполнения запроса 1.....	4
2.2 Индексы.....	4
2.3 EXPLAIN ANALYZE.....	5
3. Запрос 2.....	5
3.1 Планы выполнения запроса 2.....	6
3.2 Индексы.....	6
3.3 EXPLAIN ANALYZE.....	7

1. Задание

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. 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_ОЦЕНКИ, Н_ВЕДОМОСТИ.

Вывести атрибуты: Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

а) Н_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'неудовлетворительно'.

б) Н_ВЕДОМОСТИ.ИД < 1490007.

Вид соединения: RIGHT 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.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н_ЛЮДИ.ИД = 100012.

б) Н_ВЕДОМОСТИ.ДАТА < 2022-06-08.

в) Н_СЕССИЯ.ДАТА > 2004-01-17.

Вид соединения: LEFT JOIN.

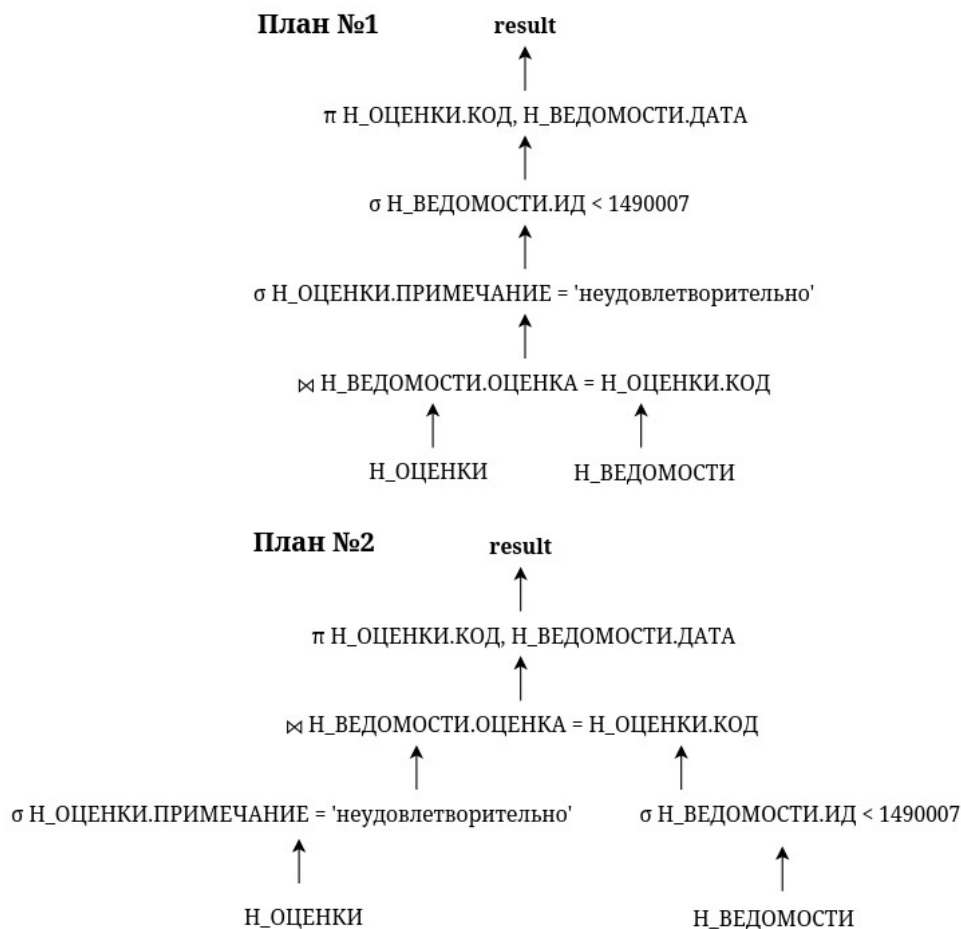
2. Запрос 1

```
SELECT Н_ОЦЕНКИ.КОД, Н_ВЕДОМОСТИ.ДАТА FROM
       Н_ОЦЕНКИ RIGHT JOIN Н_ВЕДОМОСТИ
       ON Н_ОЦЕНКИ.КОД = Н_ВЕДОМОСТИ.ОЦЕНКА
WHERE
       Н_ОЦЕНКИ.ПРИМЕЧАНИЕ < 'неудавлетворительно'
       AND Н_ВЕДОМОСТИ.ИД < 1490007;
```

```
[s467233@helios ~/databases/lab3]$ psql -h db -d ucheb -f script1.sql
```

КОД	ДАТА
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00
зачет	2010-05-27 00:00:00

2.1 Планы выполнения запроса 1



Оптимальным является план №2, так как он производит объединение таблиц по уже выбранным ранее атрибутам, а не по таблицам целиком.

2.2 Индексы

CREATE INDEX "ИНД_ВЕДОМОСТИ_ИД" ON "H_ВЕДОМОСТИ" USING BTREE("ИД");
CREATE INDEX "ИНД_ОЦЕНКИ_ПРИМЕЧАНИЕ" ON "H_ОЦЕНКИ" USING HASH("ПРИМЕЧАНИЕ");
Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. При этом для операции равенства оптимально использование HASH.

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения, а также теперь соединение таблиц будет происходить быстрее. При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее.

2.3 EXPLAIN ANALYZE

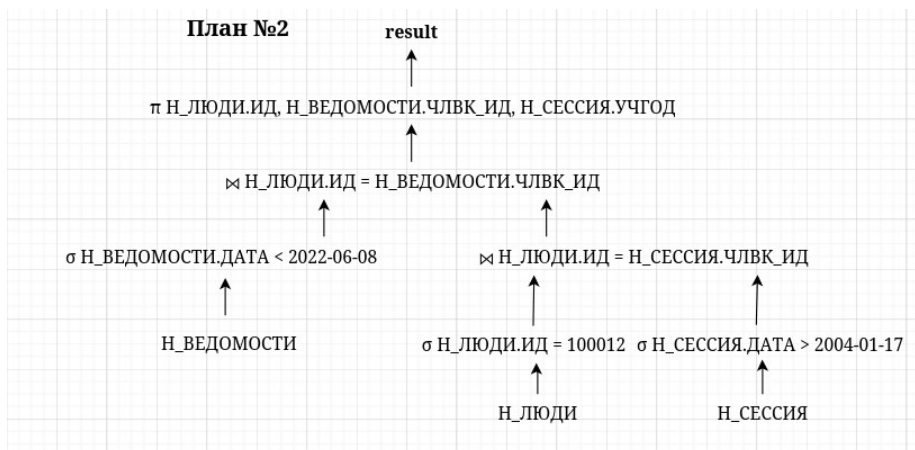
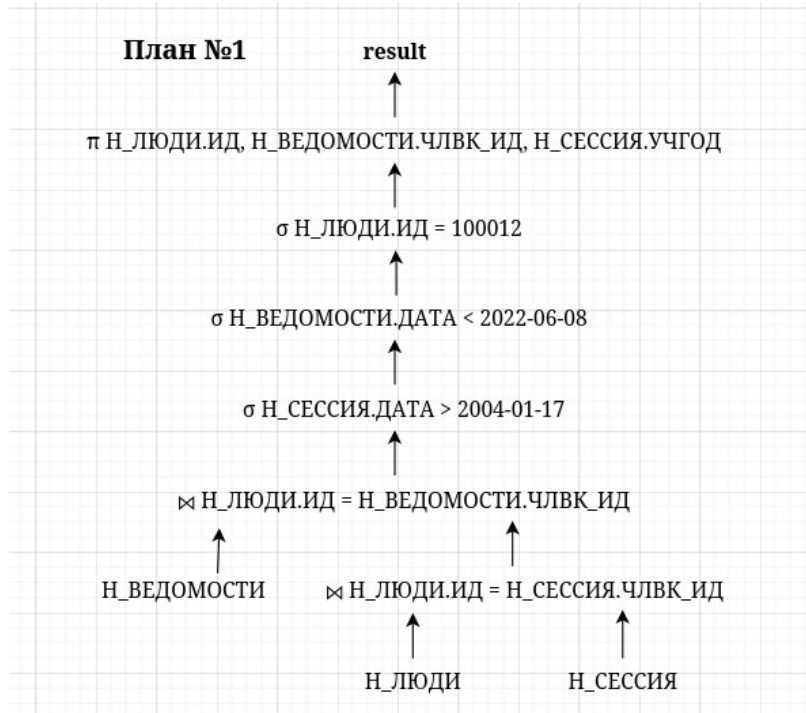
```
[s467233@helios ~/databases/lab3]$ psql -h db -d ucheb -f script1.sql
QUERY PLAN
-----
Hash Join (cost=1.15..7744.51 rows=74146 width=42) (actual time=0.109..104.700 rows=107612 loops=1)
  Hash Cond: (("Н_ВЕДОМОСТИ"."ОЦЕНКА")::text = ("Н_ОЦЕНКИ"."КОД")::text)
  -> Seq Scan on "Н_ВЕДОМОСТИ" (cost=0.00..6884.50 rows=222439 width=14) (actual time=0.012..50.907 rows=222439 loops=1)
    Filter: ("ИД" < 1490007)
    Rows Removed by Filter: 1
  -> Hash (cost=1.11..1.11 rows=3 width=34) (actual time=0.056..0.058 rows=3 loops=1)
    Buckets: 1024 Batches: 1 Memory Usage: 9kB
    -> Seq Scan on "Н_ОЦЕНКИ" (cost=0.00..1.11 rows=3 width=34) (actual time=0.037..0.045 rows=3 loops=1)
      Filter: (("ПРИМЕЧАНИЕ")::text < 'неудовлетворительно'::text)
      Rows Removed by Filter: 6
Planning Time: 1.415 ms
Execution Time: 110.363 ms
(12 строк)
```

3. Запрос 2

```
SELECT Н_ЛЮДИ.ИД, Н_ВЕДОМОСТИ.ЧЛВК_ИД, Н_СЕССИЯ.УЧГОД FROM
      Н_ЛЮДИ LEFT JOIN Н_ВЕДОМОСТИ
      ON Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
LEFT JOIN Н_СЕССИЯ
      ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE
      Н_ЛЮДИ.ИД = 100012
      AND Н_ВЕДОМОСТИ.ДАТА < '2022-06-08'
      AND Н_СЕССИЯ.ДАТА > '2004-01-17';
```

```
[s467233@helios ~/databases/lab3]$ psql -h db -d ucheb -f script2.sql
ИД | ЧЛВК_ИД | УЧГОД
----+-----+-----
(0 строк)
```

3.1 Планы выполнения запроса 2



Второй план является оптимальным. Из-за того, что выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

3.2 Индексы

```
CREATE INDEX "ИНД_СЕССИЯ_ЧЛВК_ИД" ON "Н_СЕССИЯ" USING BTREE("ЧЛВК_ИД");
CREATE INDEX "ИНД_ВЕДОМОСТИ_ДАТА" ON "Н_ВЕДОМОСТИ" USING BTREE("ДАТА");
CREATE INDEX "ИНД_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING HASH("ИД");
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование B-TREE. При этом для операции равенства оптимально использование HASH.

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения, а также теперь соединение таблиц будет происходить быстрее. При добавлении индексов планы выполнения запросов изменятся, так как вместо полного сканирования таблиц будет производиться индексный скан и Nested Loops Join станет быстрее.

3.3 EXPLAIN ANALYZE

```
[s467233@helios ~/databases/lab3]$ psql -h db -d ucheb -f script2.sql
                                QUERY PLAN
-----
Nested Loop (cost=4.90..230.58 rows=325 width=18) (actual time=0.007..0.008 rows=0 loops=1)
->  Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.29..198.97 rows=65 width=4) (actual time=0.006..0.006 rows=0 loops=1)
    Index Cond: ("ЧЛВК_ИД" = 100012)
    Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)
->  Materialize (cost=4.61..27.56 rows=5 width=14) (never executed)
    ->  Nested Loop (cost=4.61..27.53 rows=5 width=14) (never executed)
        ->  Index Only Scan using "ЧЛВК_РК" on "Н_ЛЮДИ" (cost=0.28..4.30 rows=1 width=4) (never executed)
            Index Cond: ("ИД" = 100012)
            Heap Fetches: 0
        ->  Bitmap Heap Scan on "Н_СЕССИЯ" (cost=4.33..23.18 rows=5 width=14) (never executed)
            Recheck Cond: ("ЧЛВК_ИД" = 100012)
            Filter: ("ДАТА" > '2004-01-17 00:00:00'::timestamp without time zone)
            ->  Bitmap Index Scan on "SYS_C003500_IFK" (cost=0.00..4.33 rows=6 width=0) (never executed)
                Index Cond: ("ЧЛВК_ИД" = 100012)

Planning Time: 1.087 ms
Execution Time: 0.106 ms
(16 строк)
```