**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.  
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).

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИМЯ = Александр.  
   b) Н\_ВЕДОМОСТИ.ДАТА > 2022-06-08.  
   c) Н\_ВЕДОМОСТИ.ДАТА > 2010-06-18.  
   Вид соединения: INNER JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ИД = 142095.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590.  
   Вид соединения: LEFT JOIN.

**Изначальный запрос 1**

SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ИД

FROM Н\_ЛЮДИ

INNER JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИМЯ = 'Александр'

AND Н\_ВЕДОМОСТИ.ДАТА > '2022-06-08'

AND Н\_ВЕДОМОСТИ.ДАТА > '2010-06-08';

**Изначальный запрос 2**

SELECT Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО

FROM Н\_ЛЮДИ

LEFT JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД

LEFT JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИД = 142095 AND Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД = 105590;

**Возможные индексы**

CREATE INDEX idx\_ЛЮДИ\_ИМЯ ON Н\_ЛЮДИ (ИМЯ);

CREATE INDEX idx\_ВЕДОМОСТИ\_ДАТА ON Н\_ВЕДОМОСТИ (ДАТА);

CREATE INDEX idx\_ВЕДОМОСТИ\_ЧЛВК\_ИД ON Н\_ВЕДОМОСТИ (ЧЛВК\_ИД);

Для моего запроса наиболее эффективно будет использовать B-tree индексы, так как они эффективно поддерживают равенственные и диапазонные запросы.

CREATE INDEX idx\_ЛЮДИ\_ИД ON Н\_ЛЮДИ (ИД) USING HASH;

CREATE INDEX idx\_ОБУЧЕНИЯ\_ЧЛВК\_ИД ON Н\_ОБУЧЕНИЯ (ЧЛВК\_ИД);

CREATE INDEX idx\_УЧЕНИКИ\_ЧЛВК\_ИД ON Н\_УЧЕНИКИ (ЧЛВК\_ИД);

На Н\_ЛЮДИ.ИД и Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД можно также использовать Hash, тк он эффективен для проверки условий на равенство.

**EXPLAIN ANALYZE**

**Запрос 1**

1. **Не используя индексы**

Nested Loop (cost=1000.00..6690.22 rows=1 width=20) (actual time=82.412..99.368 rows=0 loops=1)

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

" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..163.97 rows=382 width=20) (actual time=0.032..1.169 rows=382 loops=1)"

" Filter: ((""ИМЯ"")::text = 'Александр'::text)"

Rows Removed by Filter: 4736

-> Materialize (cost=1000.00..6457.51 rows=12 width=8) (actual time=0.213..0.257 rows=0 loops=382)

-> Gather (cost=1000.00..6457.45 rows=12 width=8) (actual time=81.139..98.093 rows=0 loops=1)

Workers Planned: 2

Workers Launched: 2

" -> Parallel Seq Scan on ""Н\_ВЕДОМОСТИ"" (cost=0.00..5456.25 rows=5 width=8) (actual time=17.630..17.630 rows=0 loops=3)"

" Filter: ((""ДАТА"" > '2022-06-08 00:00:00'::timestamp without time zone) AND (""ДАТА"" > '2010-06-08 00:00:00'::timestamp without time zone))"

Rows Removed by Filter: 74147

Planning Time: 0.692 ms

Execution Time: 99.417 ms

1. **Используя индекс**

Hash Join (cost=14.43..120.65 rows=1 width=20) (actual time=0.019..0.021 rows=0 loops=1)

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

" -> Bitmap Heap Scan on ""Н\_ЛЮДИ"" (cost=7.24..112.02 rows=382 width=20) (never executed)"

" Recheck Cond: ((""ИМЯ"")::text = 'Александр'::text)"

" -> Bitmap Index Scan on ""idx\_ЛЮДИ\_ИМЯ"" (cost=0.00..7.15 rows=382 width=0) (never executed)"

" Index Cond: ((""ИМЯ"")::text = 'Александр'::text)"

-> Hash (cost=7.18..7.18 rows=1 width=8) (actual time=0.009..0.009 rows=0 loops=1)

Buckets: 1024 Batches: 1 Memory Usage: 8kB

" -> Index Scan using ""idx\_ВЕДОМОСТИ\_ДАТА"" on ""Н\_ВЕДОМОСТИ"" (cost=0.29..7.18 rows=1 width=8) (actual time=0.008..0.008 rows=0 loops=1)"

" Index Cond: (""ДАТА"" > '2022-06-08 00:00:00'::timestamp without time zone)"

Planning Time: 0.855 ms

Execution Time: 0.071 ms

**Запрос 2**

1. **Не используя индексы**

Result (cost=0.00..0.00 rows=0 width=32) (actual time=0.001..0.001 rows=0 loops=1)

Planning Time: 0.316 ms

Execution Time: 0.021 ms

One-Time Filter: false

1. **Используя индексы**

Result (cost=0.00..0.00 rows=0 width=32) (actual time=0.001..0.001 rows=0 loops=1)

One-Time Filter: false

Planning Time: 1.087 ms

Execution Time: 0.016 ms

**Планы к запросу 1**

**σ**(ДАТА > '2010-06-08') Н\_ВЕДОМОМОСТИ

**σ**(ДАТА > '2022-06-08') Н\_ВЕДОМОМОСТИ

**σ**(ИМЯ = Александр) Н\_ЛЮДИ

Н\_ЛЮДИ

Н\_ВЕДОМОСТИ

**π**(Н\_ВЕДОМОСТИ.ИД, Н\_ЛЮДИ.ФАМИЛИЯ)

Н\_ЛЮДИ **⋈**

(Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД) Н\_ВЕДОМОСТИ

Н\_ЛЮДИ

Н\_ВЕДОМОСТИ

Н\_ЛЮДИ **⋈**

(Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД) Н\_ВЕДОМОСТИ

**σ**(ИМЯ = Александр) Н\_ЛЮДИ

**σ**(ДАТА > '2022-06-08') Н\_ВЕДОМОМОСТИ

**π**(Н\_ВЕДОМОСТИ.ИД, Н\_ЛЮДИ.ФАМИЛИЯ)

**EXPLAIN ANALYZE**

**SELECT Н\_ЛЮДИ.ФАМИЛИЯ, t1.ИД**

**FROM Н\_ЛЮДИ**

**INNER JOIN (**

**SELECT \* FROM temp\_ведомости**

**UNION ALL**

**SELECT \* FROM temp\_ведомости**

**UNION ALL**

**SELECT \* FROM temp\_ведомости**

**UNION ALL**

**SELECT \* FROM temp\_ведомости**

**UNION ALL**

**SELECT \* FROM temp\_ведомости**

**) AS t1 ON Н\_ЛЮДИ.ИД = t1.ЧЛВК\_ИД**

**WHERE Н\_ЛЮДИ.ИМЯ = 'Александр';**

**"Hash Join (cost=168.75..226.55 rows=30 width=20) (actual time=0.015..0.016 rows=0 loops=1)"**

**" Hash Cond: (""temp\_ведомости"".""ЧЛВК\_ИД"" = ""Н\_ЛЮДИ"".""ИД"")"**

**" -> Append (cost=0.00..56.00 rows=400 width=8) (actual time=0.015..0.015 rows=0 loops=1)"**

**" -> Seq Scan on ""temp\_ведомости"" (cost=0.00..10.80 rows=80 width=8) (actual time=0.005..0.006 rows=0 loops=1)"**

**" -> Seq Scan on ""temp\_ведомости"" ""temp\_ведомости\_1"" (cost=0.00..10.80 rows=80 width=8) (actual time=0.002..0.002 rows=0 loops=1)"**

**" -> Seq Scan on ""temp\_ведомости"" ""temp\_ведомости\_2"" (cost=0.00..10.80 rows=80 width=8) (actual time=0.002..0.002 rows=0 loops=1)"**

**" -> Seq Scan on ""temp\_ведомости"" ""temp\_ведомости\_3"" (cost=0.00..10.80 rows=80 width=8) (actual time=0.002..0.002 rows=0 loops=1)"**

**" -> Seq Scan on ""temp\_ведомости"" ""temp\_ведомости\_4"" (cost=0.00..10.80 rows=80 width=8) (actual time=0.002..0.002 rows=0 loops=1)"**

**" -> Hash (cost=163.97..163.97 rows=382 width=20) (never executed)"**

**" -> Seq Scan on ""Н\_ЛЮДИ"" (cost=0.00..163.97 rows=382 width=20) (never executed)"**

**" Filter: ((""ИМЯ"")::text = 'Александр'::text)"**

**"Planning Time: 0.325 ms"**

**"Execution Time: 0.040 ms"**