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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4

По дисциплине

“Базы данных”

Вариант 81876

Выполнил:

Ахроров Кароматуллохон Фирдавсович

Группа P3110

Проверил:

Гаврилов Антон Валерьевич

Преподаватель практики

Санкт-Петербург 2025

**1. Текст задания**

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).

**2. Реализация запросов на SQL**

-- 1. Получить фамилию и дату из Н\_ЛЮДИ и Н\_ВЕДОМОСТИ с фильтрами:

SELECT l.ФАМИЛИЯ, v.ДАТА

FROM Н\_ЛЮДИ l

JOIN Н\_ВЕДОМОСТИ v ON v.ЧЛВК\_ИД = l.ИД

WHERE l.ИД < 152862

AND v.ЧЛВК\_ИД < 142390;

-- 2. Получить отчество, дату и id сессии с фильтрами:

SELECT l.ОТЧЕСТВО, v.ДАТА, s.ИД

FROM Н\_ЛЮДИ l

LEFT JOIN Н\_ВЕДОМОСТИ v ON v.ЧЛВК\_ИД = l.ИД AND v.ИД < 1250972

LEFT JOIN Н\_СЕССИЯ s ON s.ЧЛВК\_ИД = l.ИД AND s.ЧЛВК\_ИД < 106059

WHERE l.ОТЧЕСТВО = 'Сергеевич';

**3. Уменьшение времени выполнения 1 запроса**

**Рекомендуемые индексы:**

* На Н\_ВЕДОМОСТИ(ЧЛВК\_ИД) (**B-tree**): ускоряет соединение и фильтрацию по ЧЛВК\_ИД.
* На Н\_ЛЮДИ(ИД) (**B-tree**): для соединения.

**Почему полезно:**

* Индекс ускоряет поиск нужных строк по условию < и соединение по ключу.

**Возможные планы выполнения без индексов:**  
План 1:

* Полный скан Н\_ЛЮДИ с фильтром < 152862.
* Полный скан Н\_ВЕДОМОСТИ с фильтром < 142390.
* Соединение через Nested Loop Join по ЧЛВК\_ИД.

План 2:

* Фильтрация в каждой таблице, затем Hash Join по ключу.

**Оптимальный план:**  
План 2 — фильтрация до соединения сокращает число обрабатываемых строк.

**План после добавления индексов:**  
Вместо Seq Scan — Index Scan или Index Only Scan, соединение быстрее.

π (l.ФАМИЛИЯ, v.ДАТА)

|

⋈ (v.ЧЛВК\_ИД = l.ИД)

/ \

σ (v.ЧЛВК\_ИД < 142390) σ (l.ИД < 152862)

| |

Н\_ВЕДОМОСТИ Н\_ЛЮДИ

**4. EXPLAIN ANALYZE: план 1 запроса (пример)**

"Hash Join (cost=217.70..7567.42 rows=148707 width=24) (actual time=1.661..89.086 rows=176718 loops=1)"

" Hash Cond: (v.""ЧЛВК\_ИД"" = l.""ИД"")"

" -> Seq Scan on ""Н\_ВЕДОМОСТИ"" v (cost=0.00..6884.50 rows=177078 width=12) (actual time=0.010..42.575 rows=176718 loops=1)"

" Filter: (""ЧЛВК\_ИД"" < 142390)"

" Rows Removed by Filter: 45722"

" -> Hash (cost=163.97..163.97 rows=4298 width=20) (actual time=1.634..1.637 rows=4290 loops=1)"

" Buckets: 8192 Batches: 1 Memory Usage: 291kB"

" -> Seq Scan on ""Н\_ЛЮДИ"" l (cost=0.00..163.97 rows=4298 width=20) (actual time=0.004..0.927 rows=4290 loops=1)"

" Filter: (""ИД"" < 152862)"

" Rows Removed by Filter: 828"

"Planning Time: 1.088 ms"

"Execution Time: 97.267 ms"

result

|

Hash Join

/ \

σ v."ЧЛВК\_ИД" < 142390 Hash

| |

Н\_ВЕДОМОСТИ σ l."ИД" < 152862

|

Н\_ЛЮДИ

**5. Уменьшение времени выполнения 2 запроса**

**Рекомендуемые индексы:**

* На Н\_ЛЮДИ(ОТЧЕСТВО, ИД) (**B-tree**): ускоряет фильтр и соединение.
* На Н\_СЕССИЯ(ЧЛВК\_ИД) (**B-tree**): ускоряет соединение.
* На Н\_ВЕДОМОСТИ(ЧЛВК\_ИД, ИД) (**B-tree**): ускоряет соединение и фильтр.

**Почему полезно:**

* Поиск по высокоселективным условиям и соединения выполняются быстрее.

**Планы без индексов:**  
План 1:

* Полные сканы всех таблиц по фильтрам.
* Соединение Nested Loop.

План 2:

* Сначала фильтрация, потом Hash Right Join.

**Оптимальный план:**  
План 1 — если данных немного; иначе Hash Right Join (План 2).

**План после добавления индексов:**  
Index Scan или Index Only Scan ускоряют работу.

π (l.ОТЧЕСТВО, v.ДАТА, s.ИД)

|

⟕ (LEFT JOIN по l.ИД = v.ЧЛВК\_ИД, v.ИД < 1250972)

/ \

⟕ (LEFT JOIN по l.ИД = s.ЧЛВК\_ИД, s.ЧЛВК\_ИД < 106059) σ (v.ИД < 1250972)

/ \ |

σ (l.ОТЧЕСТВО = 'Сергеевич') σ (s.ЧЛВК\_ИД < 106059) Н\_ВЕДОМОСТИ

| |

Н\_ЛЮДИ Н\_СЕССИЯ

**6. EXPLAIN ANALYZE: план 2 запроса (пример)**

"Hash Right Join (cost=298.88..8069.47 rows=15092 width=32) (actual time=1.780..55.003 rows=15216 loops=1)"

" Hash Cond: (v.""ЧЛВК\_ИД"" = l.""ИД"")"

" -> Seq Scan on ""Н\_ВЕДОМОСТИ"" v (cost=0.00..6884.50 rows=196044 width=12) (actual time=0.006..29.423 rows=196305 loops=1)"

" Filter: (""ИД"" < 1250972)"

" Rows Removed by Filter: 26135"

" -> Hash (cost=293.96..293.96 rows=394 width=28) (actual time=1.764..1.767 rows=768 loops=1)"

" Buckets: 1024 Batches: 1 Memory Usage: 52kB"

" -> Hash Right Join (cost=168.90..293.96 rows=394 width=28) (actual time=0.789..1.641 rows=768 loops=1)"

" Hash Cond: (s.""ЧЛВК\_ИД"" = l.""ИД"")"

" -> Seq Scan on ""Н\_СЕССИЯ"" s (cost=0.00..117.90 rows=2726 width=8) (actual time=0.005..0.515 rows=2728 loops=1)"

" Filter: (""ЧЛВК\_ИД"" < 106059)"

" Rows Removed by Filter: 1024"

" -> Hash (cost=163.97..163.97 rows=394 width=24) (actual time=0.757..0.759 rows=394 loops=1)"

" Buckets: 1024 Batches: 1 Memory Usage: 30kB"

" -> Seq Scan on ""Н\_ЛЮДИ"" l (cost=0.00..163.97 rows=394 width=24) (actual time=0.012..0.690 rows=394 loops=1)"

" Filter: ((""ОТЧЕСТВО"")::text = 'Сергеевич'::text)"

" Rows Removed by Filter: 4724"

"Planning Time: 0.558 ms"

"Execution Time: 55.744 ms"

result

|

Hash Right Join (v."ЧЛВК\_ИД" = l."ИД")

/ \

Seq Scan v Hash (ниже)

|

Hash Right Join (s."ЧЛВК\_ИД" = l."ИД")

/ \

Seq Scan s Hash

|

Seq Scan l

**7. Вывод**

В ходе работы я научился анализировать планы выполнения SQL-запросов, выбирать оптимальные стратегии соединения и предлагать индексы для ускорения работы. Планы EXPLAIN показали, что индексация существенно сокращает время обработки данных.