Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 4**

По дисциплине

Базы данных

Вариант №313102

*Выполнил:*

Студент группы P3131

Нодири Хисравхон

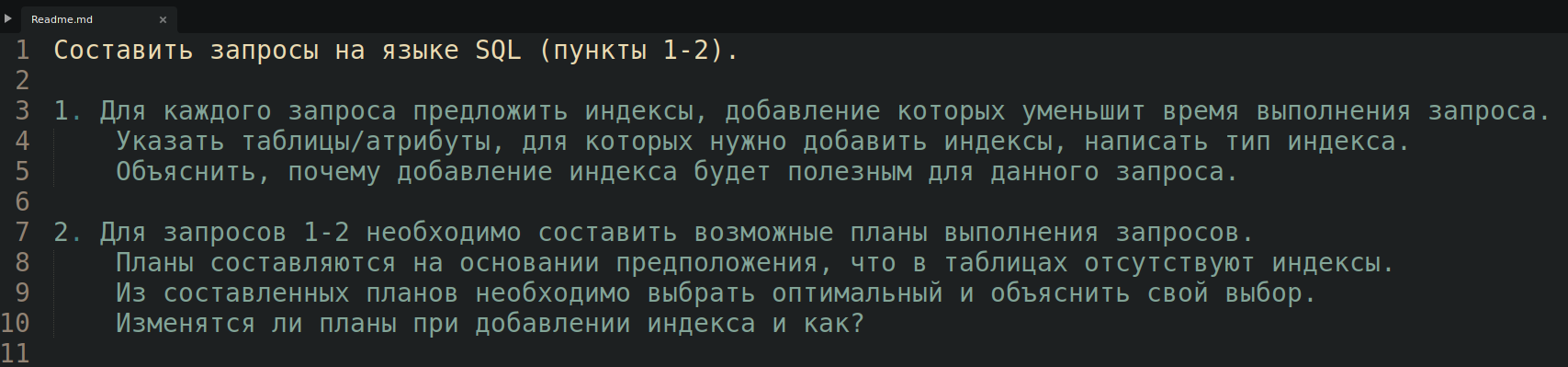
*Преподаватель:*

Наумова Надежда Александровна



Санкт-Петербург, 2023

1. **Текст задания**

**Реализация запросов на SQL**

|  |
| --- |
| *----------------------------------------------------------------------------------------------------------*  *-- 1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:*  *-- Таблицы: Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.*  *-- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.*  *-- Фильтры (AND):*  *-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2.*  *-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.*  *-- c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 117219.*  *-- Вид соединения: INNER JOIN.*  *----------------------------------------------------------------------------------------------------------*  **SELECT**  Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,  Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  **FROM**  Н\_ТИПЫ\_ВЕДОМОСТЕЙ  **INNER** **JOIN**  Н\_ВЕДОМОСТИ **ON** Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ТВ\_ИД  **WHERE**  Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2 **AND**  Н\_ВЕДОМОСТИ.ЧЛВК\_ИД <= 117219;  *----------------------------------------------------------------------------------------------------------*  *-- 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.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ЧЛВК\_ИД.*  *-- Фильтры (AND):*  *-- a) Н\_ЛЮДИ.ФАМИЛИЯ > Соколов.*  *-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219.*  *-- Вид соединения: RIGHT JOIN.*  *----------------------------------------------------------------------------------------------------------*  **SELECT**  Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД **AS** ВЕДОМОСТЬ, Н\_СЕССИЯ.ЧЛВК\_ИД **AS** СЕССИЯ  **FROM**  Н\_ЛЮДИ  **RIGHT** **JOIN**  Н\_ВЕДОМОСТИ **ON** Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  **RIGHT** **JOIN**  Н\_СЕССИЯ **ON** Н\_ВЕДОМОСТИ.СЭС\_ИД = Н\_СЕССИЯ.СЭС\_ИД  **WHERE**  Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов' **AND**  Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219; |

1. **Уменьшение выполнения времени 1 запроса**

|  |
| --- |
| Индексы, которые уменьшат время выполнения запроса:  a) На таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ:   * Индекс на атрибуте ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2" и поможет при соединении таблиц.   b) На таблице Н\_ВЕДОМОСТИ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД <= 117219". * Индекс на атрибуте ТВ\_ИД (B-tree). Это также ускорит соединение таблиц по атрибуту ТВ\_ИД.   Возможные планы выполнения запросов без индексов: План 1:   * Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра по условию "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2". * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД <= 117219". * Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.   План 2:   * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД <= 117219". * Полный скан таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ с применением фильтра по условию "Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД < 2". * Соединение таблиц с использованием Nested Loops Join по атрибуту ТВ\_ИД.   Оптимальный план:   * Оба плана одинаковы по эффективности, так как основное ограничение - это соединение таблиц, которое происходит после фильтрации.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного сканирования таблиц будет использоваться индексный скан. * Соединение таблиц с помощью Nested Loops Join станет быстрее благодаря индексу на атрибуте ТВ\_ИД. |

1. **План выполнения 1 запроса**

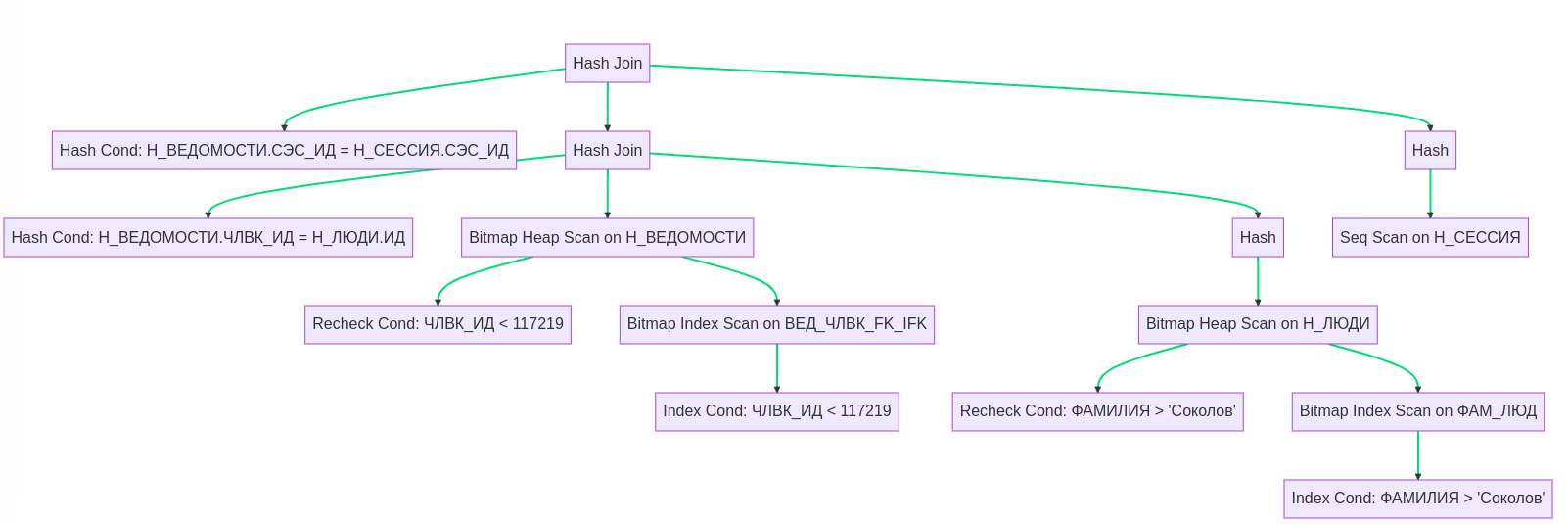
|  |
| --- |
| QUERY PLAN  ------------------------------------------------------------------------------------------------------------------------------------------  Hash Join (cost=469.87..5314.78 rows=13786 width=422) (actual time=1.865..20.937 rows=41255 loops=1)  Hash Cond: ("Н\_ВЕДОМОСТИ"."ТВ\_ИД" = "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД")  -> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=468.82..5051.79 rows=41358 width=8) (actual time=1.821..9.441 rows=42332 loops=1)  Recheck Cond: ("ЧЛВК\_ИД" <= 117219)  Heap Blocks: exact=1553  -> Bitmap Index Scan on "ВЕД\_ЧЛВК\_FK\_IFK" (cost=0.00..458.48 rows=41358 width=0) (actual time=1.609..1.610 rows=42332 loops=1)  Index Cond: ("ЧЛВК\_ИД" <= 117219)  -> Hash (cost=1.04..1.04 rows=1 width=422) (actual time=0.022..0.024 rows=1 loops=1)  Buckets: 1024 Batches: 1 Memory Usage: 9kB  -> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.015..0.018 rows=1 loops=1)  Filter: ("ИД" < 2)  Rows Removed by Filter: 2  Planning Time: 0.302 ms  Execution Time: 22.934 ms |

1. **Уменьшение выполнения времени 2 запроса**

|  |
| --- |
| Индексы, которые уменьшат время выполнения запроса:  a) На таблице Н\_ЛЮДИ:   * Индекс на атрибуте ФАМИЛИЯ (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов'". * Индекс на атрибуте ИД (B-tree). Это поможет при соединении таблиц.   b) На таблице Н\_ВЕДОМОСТИ:   * Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219" и поможет при соединении таблиц. * Индекс на атрибуте СЭС\_ИД (B-tree). Это поможет при соединении таблиц.   c) На таблице Н\_СЕССИЯ:   * Индекс на атрибуте СЭС\_ИД (B-tree). Это также поможет при соединении таблиц.   Возможные планы выполнения запросов без индексов: План 1:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов'". * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219". * Соединение таблиц Н\_ЛЮДИ и Н\_ВЕДОМОСТИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД. * Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Nested Loops Join по атрибуту СЭС\_ИД.   План 2:   * Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ФАМИЛИЯ > 'Соколов'". * Полный скан таблицы Н\_ВЕДОМОСТИ с применением фильтра по условию "Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 117219". * Соединение таблиц Н\_ЛЮДИ и Н\_ВЕДОМОСТИ с использованием Hash Join по атрибуту ЧЛВК\_ИД. * Соединение результатов предыдущего этапа с таблицей Н\_СЕССИЯ с использованием Hash Join по атрибуту СЭС\_ИД.   Оптимальный план:   * План 1, так как Nested Loops Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие, и заранее известно, что результаты фильтрации будут содержать малое количество строк.   При добавлении индексов планы выполнения запросов изменятся:   * Вместо полного скана таблиц будет использоваться индексный скан. * Nested Loops Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД и СЭС\_ИД. |

1. **План выполнения 2 запроса**

|  |
| --- |
| QUERY PLAN  ------------------------------------------------------------------------------------------------------------------------------------------------  Hash Join (cost=779.69..5600.48 rows=5325 width=21) (actual time=14.260..19.140 rows=505 loops=1)  Hash Cond: ("Н\_ВЕДОМОСТИ"."СЭС\_ИД" = "Н\_СЕССИЯ"."СЭС\_ИД")  -> Hash Join (cost=624.27..5314.93 rows=8786 width=21) (actual time=2.726..16.705 rows=9197 loops=1)  Hash Cond: ("Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" = "Н\_ЛЮДИ"."ИД")  -> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=468.32..5050.50 rows=41294 width=8) (actual time=1.763..9.104 rows=42301 loops=1)  Recheck Cond: ("ЧЛВК\_ИД" < 117219)  Heap Blocks: exact=1553  -> Bitmap Index Scan on "ВЕД\_ЧЛВК\_FK\_IFK" (cost=0.00..458.00 rows=41294 width=0) (actual time=1.556..1.556 rows=42301 loops=1)  Index Cond: ("ЧЛВК\_ИД" < 117219)  -> Hash (cost=142.33..142.33 rows=1089 width=17) (actual time=0.940..0.942 rows=1090 loops=1)  Buckets: 2048 Batches: 1 Memory Usage: 70kB  -> Bitmap Heap Scan on "Н\_ЛЮДИ" (cost=28.72..142.33 rows=1089 width=17) (actual time=0.412..0.757 rows=1090 loops=1)  Recheck Cond: (("ФАМИЛИЯ")::text > 'Соколов'::text)  Heap Blocks: exact=100  -> Bitmap Index Scan on "ФАМ\_ЛЮД" (cost=0.00..28.45 rows=1089 width=0) (actual time=0.394..0.394 rows=1090 loops=1)  Index Cond: (("ФАМИЛИЯ")::text > 'Соколов'::text)  -> Hash (cost=108.52..108.52 rows=3752 width=8) (actual time=1.255..1.256 rows=3752 loops=1)  Buckets: 4096 Batches: 1 Memory Usage: 177kB  -> Seq Scan on "Н\_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=8) (actual time=0.010..0.694 rows=3752 loops=1)  Planning Time: 0.689 ms  Execution Time: 19.233 ms |

****

1. **Вывод**

В ходе выполнения лабораторной работы я разобрался в работе с реляционной алгеброй и изучил, как строить планы выполнения запросов и их диаграммы. Также я освоил различные виды индексов и понял, как их использовать для оптимизации скорости выполнения запросов. Теперь я готов применять эти знания на практике для эффективной работы с базами данных и улучшения производительности моих SQL-запросов.