МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

“Информационные системы и базы данных”

Вариант № 312446

Выполнил: Воробьев Кирилл

Группа: P33302

Преподаватель: Шешуков Дмитрий

Михайлович

Изображение выглядит как текст, коллекция картинок

Автоматически созданное описание

Санкт-Петербург, 2022

**Задание**

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:  
   Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.  
   Фильтры (AND):  
   a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Экзаменационный лист.  
   b) 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.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ИД.  
   Фильтры (AND):  
   a) Н\_ЛЮДИ.ИМЯ = Роман.  
   b) Н\_ВЕДОМОСТИ.ИД < 1457443.  
   c) Н\_СЕССИЯ.ИД > 27640.  
   Вид соединения: RIGHT JOIN.

**Выполнение**

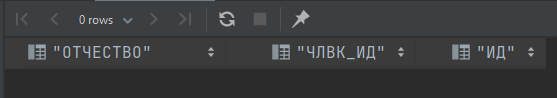
**Запросы**

1. SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД" WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" > 'Экзаменационный лист' AND "Н\_ВЕДОМОСТИ"."ИД" > 1490007

Изображение выглядит как текст

Автоматически созданное описание

1. SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД", "Н\_СЕССИЯ"."ИД" 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" = 'Роман' AND "Н\_ВЕДОМОСТИ"."ИД" < 1457443 AND "Н\_СЕССИЯ"."ИД" > 27640



**Индексы**

1. CREATE INDEX statements\_id ON "Н\_ВЕДОМОСТИ" USING btree ("ИД")

Я считаю, что для оптимизации выполнения первого запроса имеет смысл использовать B-tree индекс для атрибута "ИД" таблицы "Н\_ВЕДОМОСТИ", так как в запросе у нас производится сравнение «меньше». Для условия со сравнением "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" считаю не имеет смысла использовать индекс, поскольку возможных значений всего 3 и производится сравнение «больше»

1. CREATE INDEX session\_id ON "Н\_СЕССИЯ" USING btree ("ИД")

CREATE INDEX statements\_id ON "Н\_ВЕДОМОСТИ" USING btree("ИД")

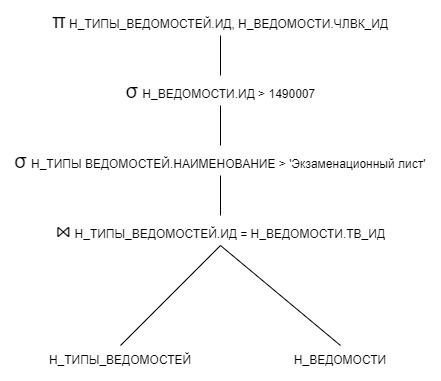
CREATE INDEX person\_name ON "Н\_ЛЮДИ" USING hash("ИМЯ")

Я считаю, что для оптимизации выполнения второго запроса имеет смысл использовать B-tree индекс для атрибута "ИД" таблицы "Н\_СЕССИЯ", так как в запросе у нас производится сравнение «больше». Также, данный индекс имеет смысл использовать для атрибута "ИД" таблицы "Н\_ВЕДОМОСТИ", так как в запросе у нас производится сравнение «меньше». А для проверки третьего условия имеет смысл использовать Hash индекс, так как производится проверка равенства строк и в этой ситуации данный индекс будет отрабатывать эффективнее.

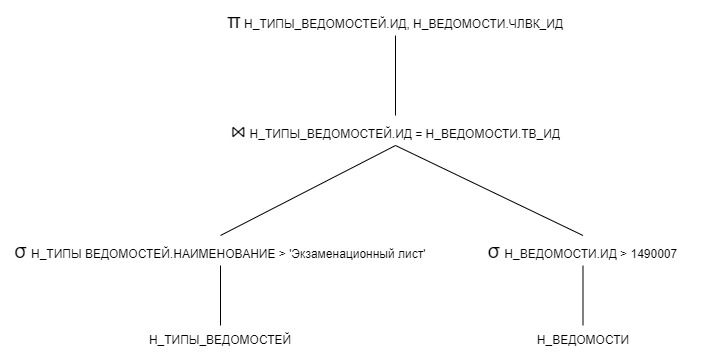
**Возможные планы выполнения запросов**

**Первый запрос**

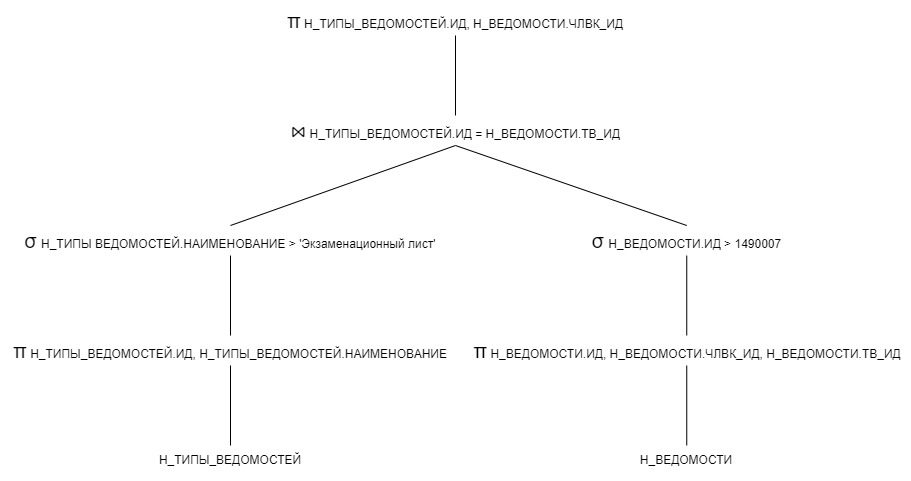
1. Соединение двух отношений, после которого производятся операции выборки по первому и второму предикатам. Результатом является проекция двух атрибутов.



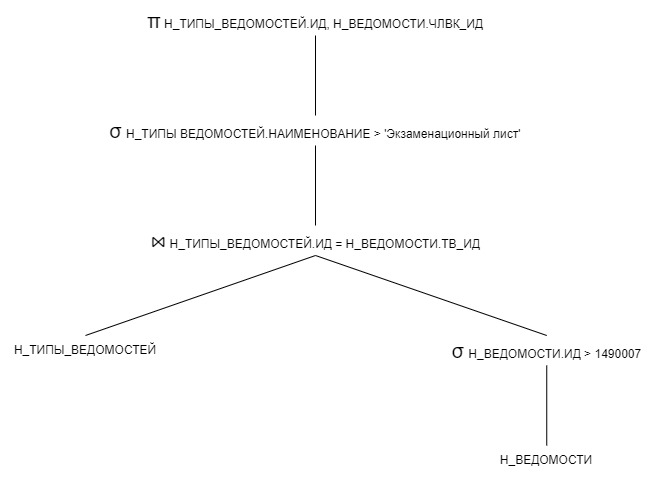
1. Изначально производятся операции выборки, далее соединение двух отношений. Результатом является проекция двух атрибутов.



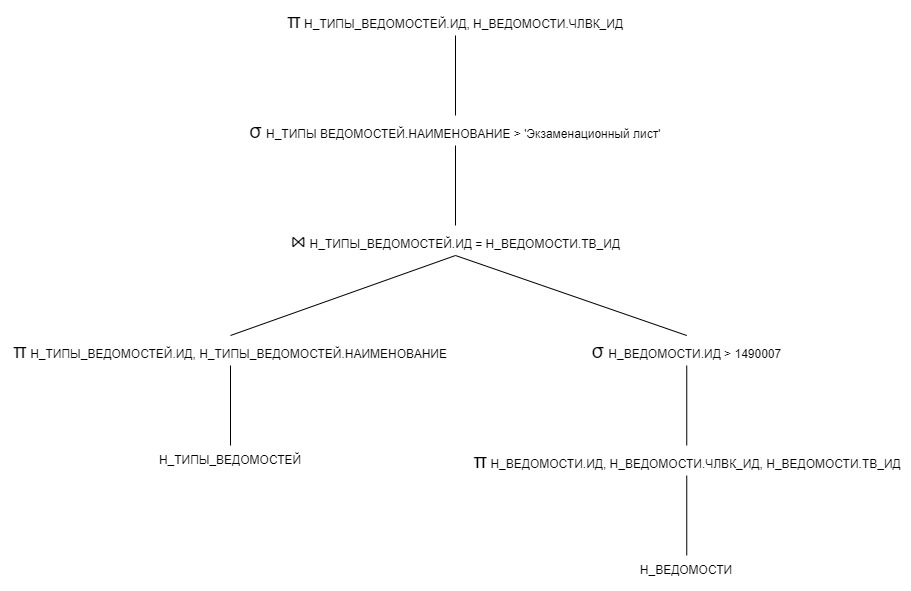
1. Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка и соединение. Результатом является проекция двух атрибутов.



1. Для начала производится выборка для одного из отношений, далее соединение двух отношений, после чего производится выборка по результату соединения. Результатом является проекция двух атрибутов



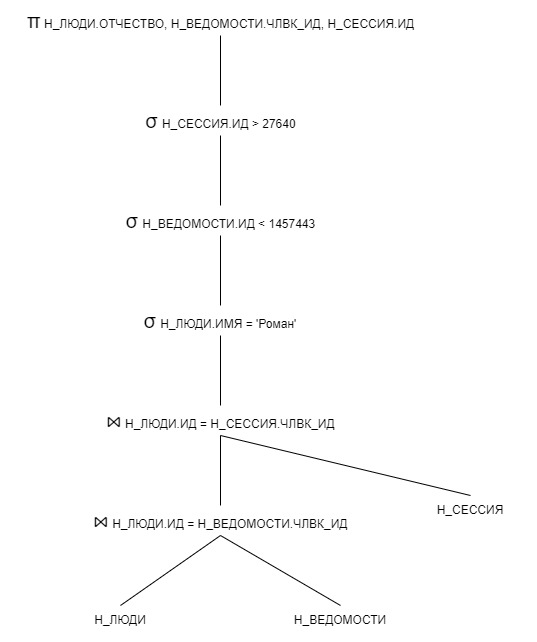
1. Для начала получается проекция для двух отношений из атрибутов, необходимых для выборки, соединения и итоговой проекции, далее производится выборка для одного из отношений, а после соединение двух отношений, после которого производится выборка для результата соединения. Результатом является проекция двух атрибутов.



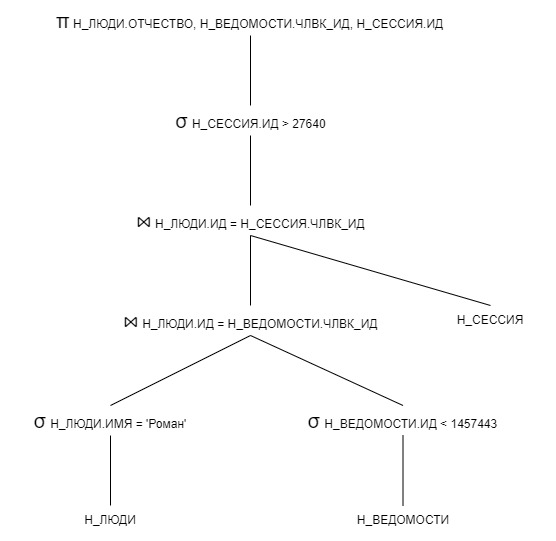
Из составленных возможных планов выполнения запроса лучшим является третий, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняется соединение. Это позволяет уменьшить размер хранимых данных.

**Второй запрос**

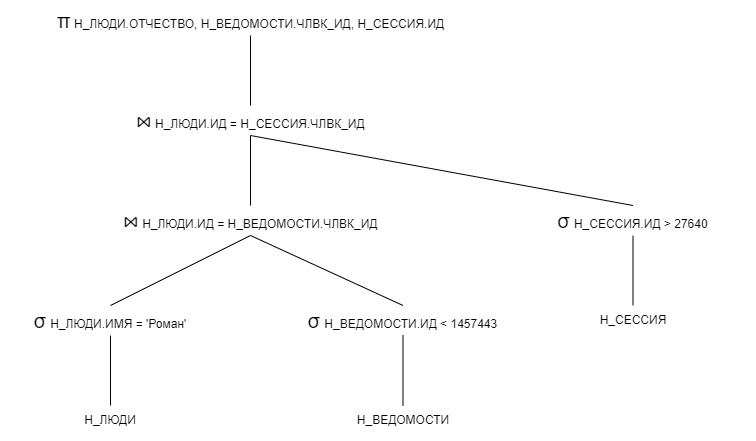
1. Соединение двух отношений, далее результат соединения соединяется с третьим отношением, а для результата двух соединений последовательно производится выборка. Результатом является проекция трех атрибутов.



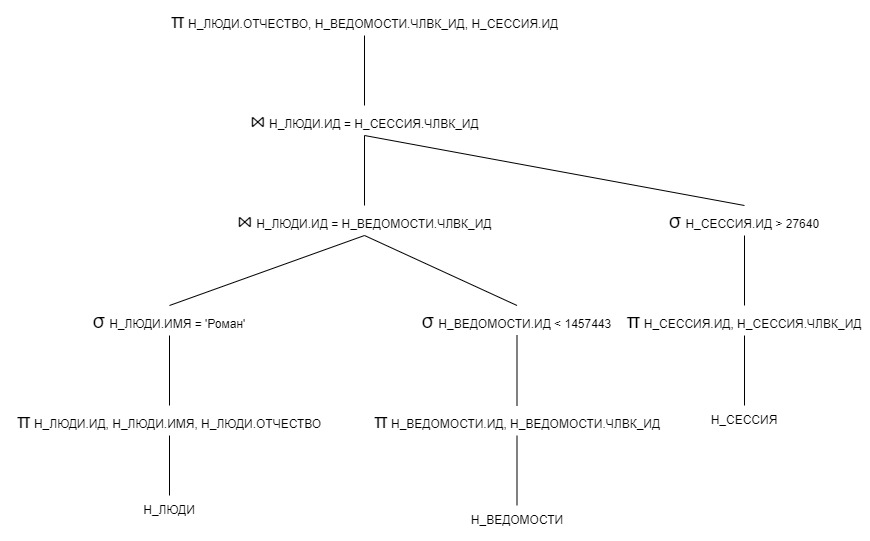
1. Для начала производится выборка для первых двух отношений, далее производится соединение этих двух отношений, а после результат соединения соединяется с третьим отношением. По полученному соединению производится выборка. Результатом является проекция трех атрибутов.



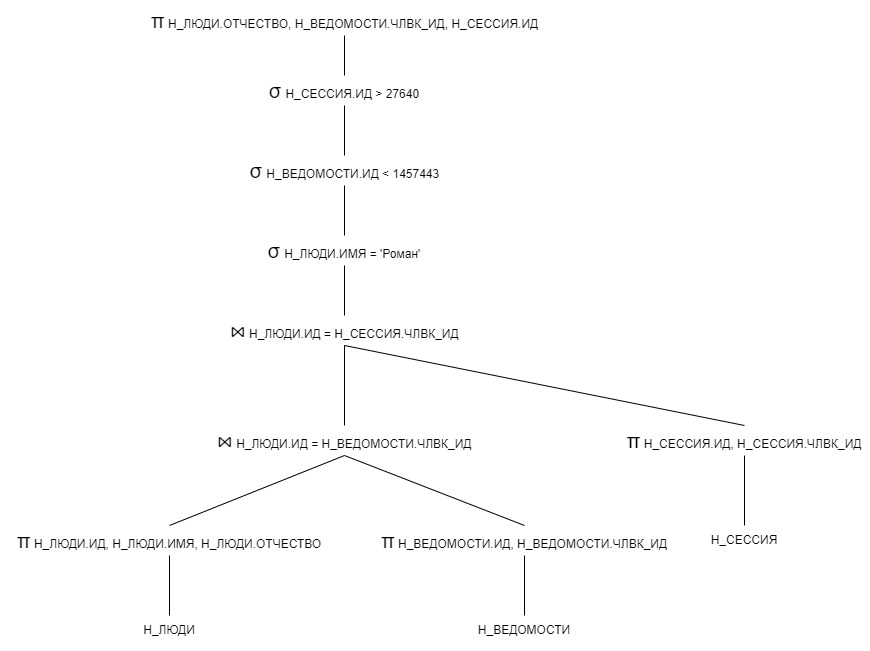
1. Для начала производится выборка для двух отношений, после соединение этих двух отношений, а далее соединение с третьим отношением, по которому перед соединением была произведена выборка. Результатом является проекция трех атрибутов.



1. Сначала получается проекция для двух отношений, после по ним производится выборка и соединение. Далее результат соединения соединяется с третьим отношением, для которого перед соединением была получена проекция и произведена выборка. Результатом является проекция трех атрибутов.



1. Получаются проекции для двух отношений, после чего эти отношения соединяются, а результат соединения соединяется с третьим отношением, для которого перед соединением тоже была получена проекция. Далее производится последовательная выборка. Результатом является проекция трех атрибутов.



Из составленных возможных планов выполнения запроса лучшим является пятый, поскольку в нем изначально получаются проекции и производятся выборки, а уже после этого выполняются соединения. Это позволяет уменьшить размер хранимых данных.

**Команда EXPLAIN ANALYZE**

1. EXPLAIN ANALYZE SELECT "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" FROM "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" RIGHT JOIN "Н\_ВЕДОМОСТИ" ON "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД" WHERE "Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" > 'Экзаменационный лист' AND "Н\_ВЕДОМОСТИ"."ИД" > 1490007 Изображение выглядит как текст

   Автоматически созданное описание
2. EXPLAIN ANALYZE SELECT "Н\_ЛЮДИ"."ОТЧЕСТВО", "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД", "Н\_СЕССИЯ"."ИД" 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" = 'Роман' AND "Н\_ВЕДОМОСТИ"."ИД" < 1457443 AND "Н\_СЕССИЯ"."ИД" > 27640 Изображение выглядит как текст

   Автоматически созданное описание

**Вывод**

В ходе выполнения данной лабораторной работы я познакомился с индексами, тем как они влияют на нагрузку на систему. Также я познакомился с планом выполнения запроса, узнал каким образом СУБД выбирает оптимальный. И еще узнал что выполняет команда EXPLAIN ANALYZE.