

Факультет программной инженерии и компьютерной техники

Информационные системы и базы данных

Лабораторная работа №3

Вариант №1754

Преподаватель: Харитонова Анастасия Евгеньевна

Выполнил: Кульбако Артемий Юрьевич Р33112

# 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.  
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).

# Запросы

1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

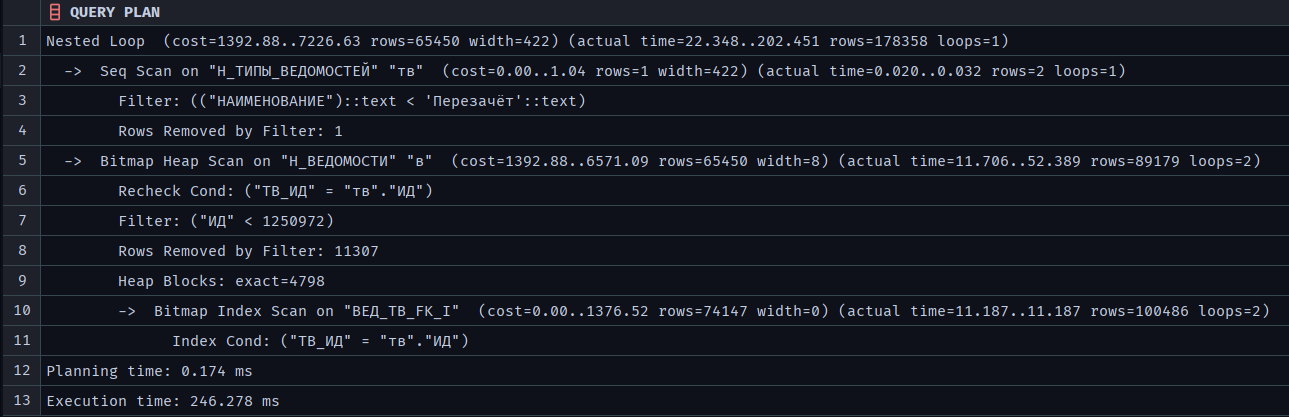
Фильтры (AND):

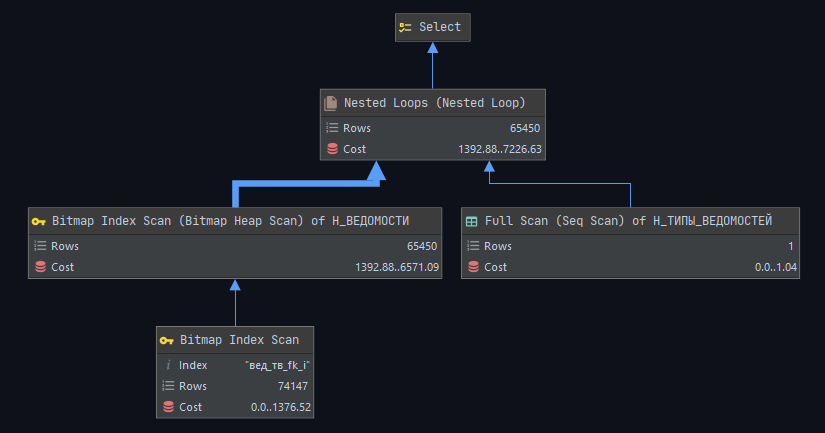
a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Перезачет.

b) Н\_ВЕДОМОСТИ.ИД < 1250972.

Вид соединения: LEFT JOIN.

*select* "НАИМЕНОВАНИЕ", "ЧЛВК\_ИД" *from* "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" тв  
 *join* "Н\_ВЕДОМОСТИ" в *on* в."ТВ\_ИД" = тв."ИД"  
*where* "НАИМЕНОВАНИЕ" < 'Перезачёт' *and* в."ИД" < 1250972;





Данные из “Н\_ВЕДОМОСТИ” отбираются по уникальному ключу таблицы “ИД”, тем не менее, индекс для этой таблицы создан по столбцу «ЧЛВК\_ИД». Сначала может показаться, что есть смысл построить индекс и по «ИД», но эта мысль ошибочна. Дело в том, что для него мы имеем 2 критерия отбора, и чтобы избежать повторных чтений строк, планировщик строит битовую карту – по одной на каждое условие, а затем объединяет их.

“Н\_ТИПЫ\_ВЕДОМОСТЕЙ” сканируется полностью, тем не менее индекс таблице не нужен, т.к. она содержит всего 3 записи и маловероятно, что когда-нибудь будет расширена: про негативное влияние от добавления индекса в малую таблицу сказано и в официальной документации к PostgreSQL (<https://postgrespro.ru/docs/postgrespro/10/using-explain>, п. 14.1.1), где приводится аналогичный пример, с двумя таблицами «Но для этого потребуется обойти оба индекса, так что это не обязательно будет выгоднее, чем просто просмотреть один индекс, а второе условие обработать как фильтр».

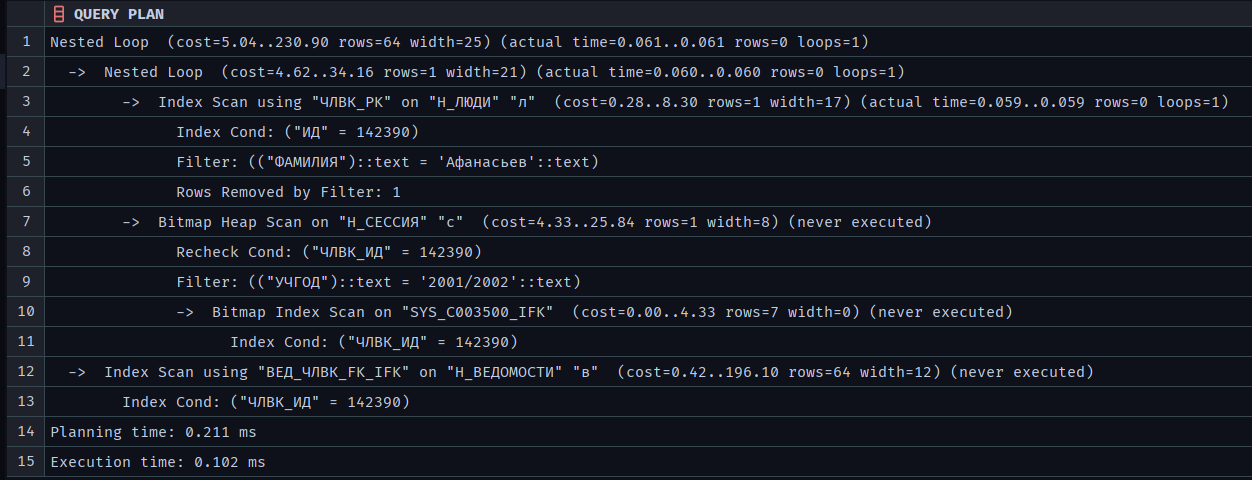
Относительно медленное последовательное сканирование выполняется только на последнем этапе, при котором отбираются строки с условием “НАИМЕНОВАНИЕ” < ‘Перезачёт’.

**Оптимизировать этот запрос не нужно.**

1. 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.ДАТА, Н\_СЕССИЯ.ИД.  
Фильтры (AND):   
a) Н\_ЛЮДИ.ФАМИЛИЯ = Афанасьев.  
b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 142390.  
c) Н\_СЕССИЯ.УЧГОД = 2001/2002.  
Вид соединения: LEFT JOIN.

*select* л."ИМЯ", в."ДАТА", с."ИД" *from* "Н\_ЛЮДИ" л  
 *join* "Н\_ВЕДОМОСТИ" в *on* в."ЧЛВК\_ИД" = л."ИД"  
 *join* "Н\_СЕССИЯ" с *on* с."ЧЛВК\_ИД" = л."ИД"  
*where* "ФАМИЛИЯ" = 'Афанасьев' *and* в."ЧЛВК\_ИД" = 142390 *and* "УЧГОД" = '2001/2002';



# 

Здесь мы имеем обращение к трём таблицам, при этом, составных условия для какого-либо из полей нет.

В “Н\_ВЕДОМОСТИ” индекс уже построен по необходимому полю.

Выборка по индексу происходит и из таблицы “Н\_ЛЮДИ”, после чего происходит фильтрация по “ФАМИЛИЯ”. Из инфологической модели мы знаем, что “ИД” человека уникально определяет человека, поэтому смысла строит индекс по фамилии в данном запросе смысла нет, мы всегда получим одну строку: больше смысла будет в отказе от одного из условий выбора (если отказать от “ИД”, то только тогда нужно строить индекс по “ФАМИЛИЯ”).

Последняя таблица – “Н\_СЕССИЯ”, где действительно стоит создать индекс по “УЧГОД”, т.к. довольно частый сценарий использования этой таблицы вообще, а также имеет смысл в данном запросе.

*create unique index* "Н\_СЕССИЯ-УЧГОД-BTREE" *on* "Н\_СЕССИЯ" *using* btree("УЧГОД");

# Вывод

В процессе выполнения лабораторной я немного научился оптимизировать БД путём создания индексов – структур данных, распределяющих записи из таблиц по заданному полю, что ускоряет обращение к ним, а также в процессе выборки данных планировщиком. Основная сложность задания – разобраться в неочевидном из-за названия процессе выборки через Bitmap-индекс.