Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Базы данных

**Лабораторная работа № 4**

Вариант № 1301

Выполнил:

Сандов Кирилл Алекссевич

Группа:

P3113

Проверил:

преподаватель практики Горбунов Михаил Витальевич

Санкт-Петербург

2023

# Оглавление

[Оглавление 2](#_Toc135430195)

[Задание 3](#_Toc135430196)

[Реализация запросов 5](#_Toc135430197)

[Запрос 1 5](#_Toc135430198)

[Запрос 2 5](#_Toc135430199)

[Применение индексов 6](#_Toc135430200)

[Запрос 1 6](#_Toc135430201)

[Запрос 2 7](#_Toc135430202)

[Планы выполнения запросов 10](#_Toc135430203)

[Запрос 1 10](#_Toc135430204)

[Запрос 2 12](#_Toc135430205)

[Анализ выбора планировщика 14](#_Toc135430207)

[Запрос 1 14](#_Toc135430208)

[Запрос 2 14](#_Toc135430209)

[Заключение 15](#_Toc135430210)

# 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).

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.ИД.

Фильтры (AND):

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

b) Н\_ВЕДОМОСТИ.ДАТА = 2022-06-08.

Вид соединения: LEFT JOIN.

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

Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ЧЛВК\_ИД.

Фильтры (AND):

a) Н\_ЛЮДИ.ФАМИЛИЯ > Ёлкин.

b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249.

c) Н\_СЕССИЯ.УЧГОД < 2008/2009.

Вид соединения: INNER JOIN.

# Реализация запросов

## **Запрос 1**

|  |
| --- |
| SELECT Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД, Н\_ВЕДОМОСТИ.ИД  FROM Н\_ВЕДОМОСТИ  LEFT JOIN Н\_ТИПЫ\_ВЕДОМОСТЕЙ ON Н\_ВЕДОМОСТИ.ТВ\_ИД = Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД  WHERE (  Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < 'Ведомость'  AND  Н\_ВЕДОМОСТИ.ДАТА = '2022-06-08'  ); |

Листинг 1

## **Запрос 2**

|  |
| --- |
| SELECT Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ВЕДОМОСТИ.ДАТА, Н\_СЕССИЯ.ЧЛВК\_ИД  FROM Н\_ЛЮДИ  JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД  JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД  WHERE (  Н\_ЛЮДИ.ФАМИЛИЯ > 'Ёлкин' AND  Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = 163249 AND  Н\_СЕССИЯ.УЧГОД < '2008/2009'  ); |

Листинг 2

# Применение индексов

Далее для каждого из запросов будут записаны возможные индексы, мотивация к их применению и оценка их эффективности.

## Запрос 1

1) Индекс на атрибут ДАТА таблицы Н\_ВЕДОМОСТИ, тип индекса – Hash.

|  |
| --- |
| CREATE INDEX Н\_ВЕДОМОСТИ\_ДАТА\_I  ON Н\_ВЕДОМОСТИ  USING hash(ДАТА) |

Листинг 3

Посмотрим на статистику данного атрибута в специальной таблице pg\_stats:

|  |
| --- |
| +----------+-----------+ |n\_distinct|correlation| +----------+-----------+ |1836 |0.5155182 | +----------+-----------+ |

Листинг 4

Мы видим, что значение n\_distinct, показывающее, сколько уникальных значений данного атрибута существует в таблице в данный момент, довольно большое, а также correlation ближе к 1, чем к 0, что означает необходимость в применение индекса по данному атрибуту. Ведь это упорядочит значения, что ускорит фильтрацию по ним в блоке WHERE.

Также был выбран тип Hash, потому что он наиболее быстрый для операции равенства.

Причины, почему другие индексы будут бесполезные для данного запроса:

* Индекс в таблице Н\_ТИПЫ\_ВЕДОМОСТЕЙ на атрибут НАИМЕНОВАНИЕ – он бесполезен, так как в данной таблице всего три строки.

## Запрос 2

1) Индекс на атрибут ЧЛВК\_ИД таблицы Н\_СЕССИЯ, тип индекса – Hash. С укзанием того, что применяем на значениях не NULL.

|  |
| --- |
| CREATE INDEX Н\_СЕССИЯ\_ЧЛВК\_ИД\_I  ON Н\_СЕССИЯ  USING hash(ЧЛВК\_ИД)  WHERE ЧЛВК\_ИД IS NOT NULL |

Листинг 5

Статистика данного атрибута:

|  |
| --- |
| +----------+----------+-----------+ |null\_frac |n\_distinct|correlation| +----------+----------+-----------+ |0.13672708|180 |0.13944401 | +----------+----------+-----------+ |

Листинг 6

Видим, что n\_distinct и correlation не говорят о том, что индекс сильно необходим. Однако вспомним, что данный атрибут участвует в операции соединения с таблицей Н\_ЛЮДИ, где строк довольно много. Поэтому к этому атрибуту будет довольного много обращений, что подтверждает полезность индекса. Также заметим, что null\_frac (доля значений NULL среди всех значений) имеет некоторое не малое значений. А при соединении вида INNER JOIN к Н\_ЛЮДИ значения NULL использоваться не будут. Следовательно, мы можем добавить оптимизацию, исключив из индексации значения NULL.

Был выбран тип индекса Hash, так как в операции соединения постоянно будут использовать сравнения по равенству.

2) Индекс на атрибут ЧЛВК\_ИД таблицы Н\_ВЕДОМОСТИ, тип индекса – Hash.

|  |
| --- |
| CREATE INDEX Н\_ВЕДОМОСТИ\_ЧЛВК\_ИД\_I  ON Н\_ВЕДОМОСТИ  USING hash(ЧЛВК\_ИД); |

Листинг 7

Статистика атрибута:

|  |
| --- |
| +---------+----------+-----------+  |null\_frac|n\_distinct|correlation|  +---------+----------+-----------+  |0 |3264 |0.5222814 |  +---------+----------+-----------+ |

Листинг 8

В данном случае мотивация совпадает с предыдущим индексом. Но мы не исключили значения NULL, потому что их нет (null\_frac равен 0).

3) Индекс на атрибуты ФАМИЛИЯ, ИД таблицы Н\_ЛЮДИ, тип индекса B-tree. С дополнительным условием выборки: ФАМИЛИЯ > 'Ёлкин'.

|  |
| --- |
| CREATE INDEX Н\_ЛЮДИ\_ФАМИЛИЯ\_ИД\_I  ON Н\_ЛЮДИ  USING btree(ФАМИЛИЯ, ИД)  WHERE ФАМИЛИЯ > 'Ёлкин'; |

Листинг 9

Статистика атрибута ФАМИЛИЯ:

|  |
| --- |
| +---------+----------+------------+  |null\_frac|n\_distinct|correlation |  +---------+----------+------------+  |0 |-0.7317311|-0.000653284|  +---------+----------+------------+ |

Листинг 10

Отрицательное значение n\_distinct означает (если взять его по модулю) долю уникальных значений среди всех значений. Это доля довольно большая, поэтому созданный индекс с фильтром отсеет большое количество ненужных запросу значений и ускорит фильтрацию. Также в состав индекса был включён атрибут ИД, поскольку при выполнении запроса нужно будет проводить фильтрацию по обоим атрибутам таблицы.

Был выбран тип B-tree, так как он поддерживает операции сравнения, а у нас как раз таки сравнение по полю ФАМИЛИЯ.

Причины, почему не были созданы другие индексы:

* Индекс в таблице Н\_СЕССИЯ на атрибут УЧГОД – при просмотре статистики данной таблицы было выявлено, что в ней всего 11 уникальных значений, что слишком мало для использования индекса;
* Индекс в таблице Н\_ЛЮДИ на атрибут ИД – он уже создан СУБД автоматически как индекс для основного ключа.

# Планы выполнения запросов

## Запрос 1

План 1

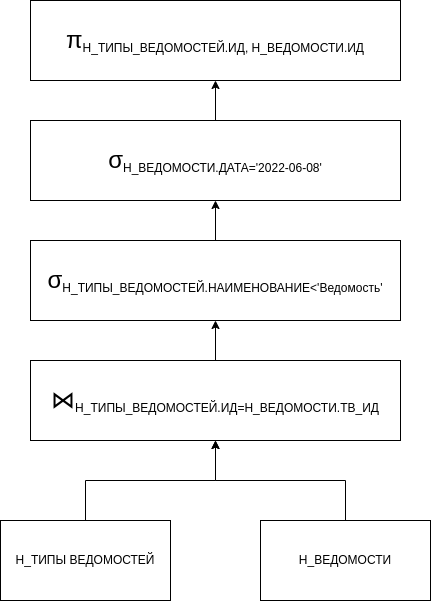


Рисунок 1

План 2

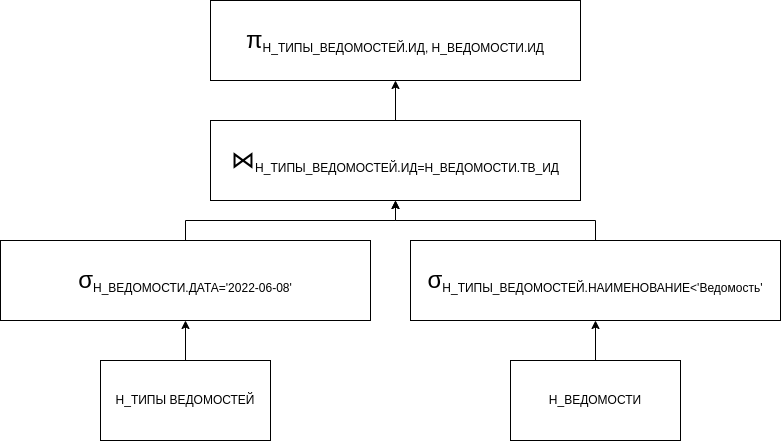


Рисунок 2

План 3

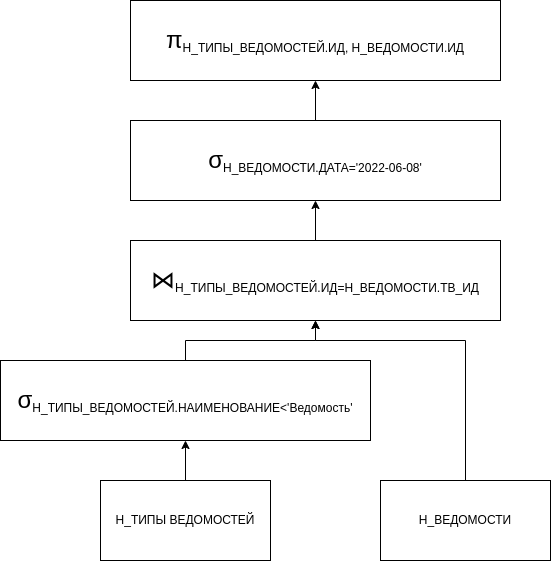


Рисунок 3

План 4

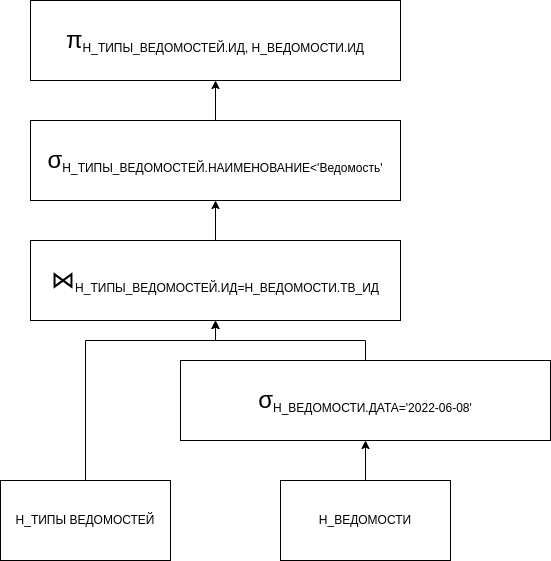


Рисунок 4

Наиболее эффективным является план 2, поскольку в нём мы сначала отфильтруем таблицы, что является довольно быстрой операцией, а потом будет соединять. В результате будет соединяться не так много данных, как могло быть, если бы мы не отфильтровали таблицы перед этим.

При добавлении индекса планы не изменятся, так как он повлияет лишь на эффективность фильтрации данных в таблице Н\_ВЕДОМОСТИ.

## Запрос 2

## **План 1**

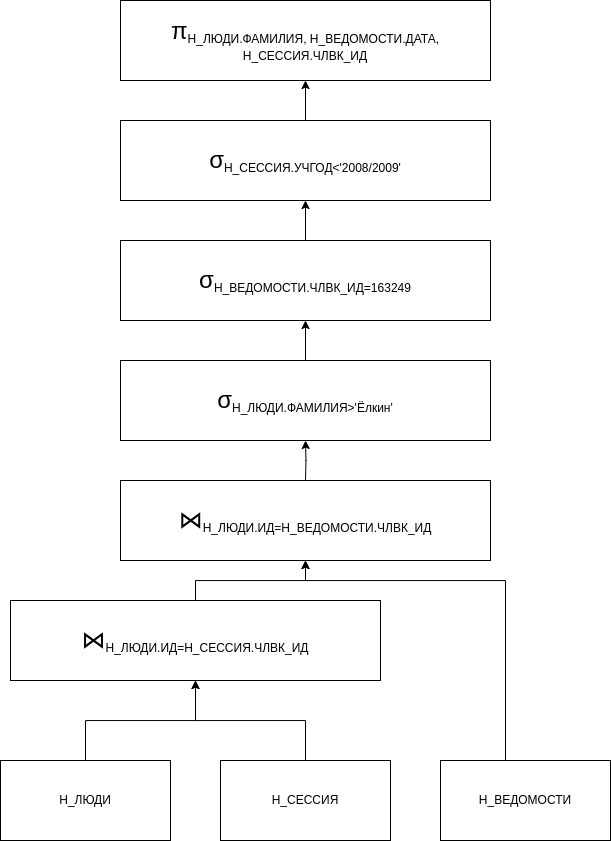


Рисунок 5

План 2

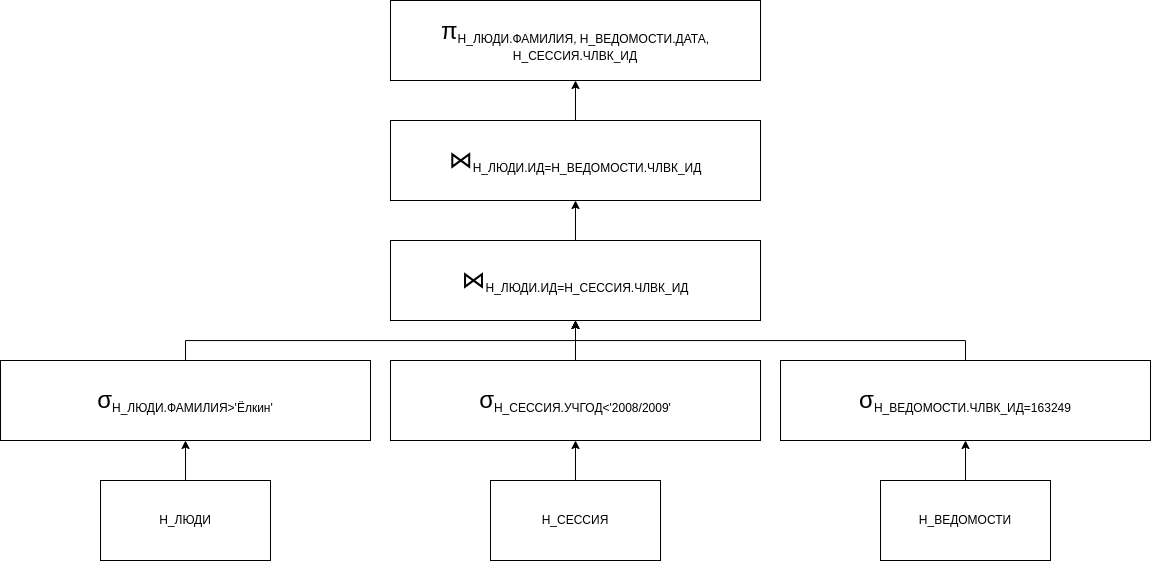


Рисунок 6

Можно составить ещё много вариантов последовательного расположения операций фильтрации и соединений, но все дальнейшие комбинации не будут эффективны, поэтому их можно не изображать. Опять же, оптимальным планом будет план 2, потому что в нём данные сначала фильтруются, а потом используются в соединении.

При добавлении индекса планировщик запроса обнаружит, что благодаря составному индексу по атрибутам ФАМИЛИЯ, ИД в таблице Н\_ЛЮДИ выгоднее сделать фильтрацию по нему, взяв за значение ИД значение внешнего ключа Н\_ВЕДОМОСТИ.ЧЛВК\_ИД. Также индекс в таблице Н\_СЕССИЯ по атрибуту ЧЛВК\_ИД будет использован для фильтрации по этому значению. Тогда план 2 будет выглядеть следующим образом:

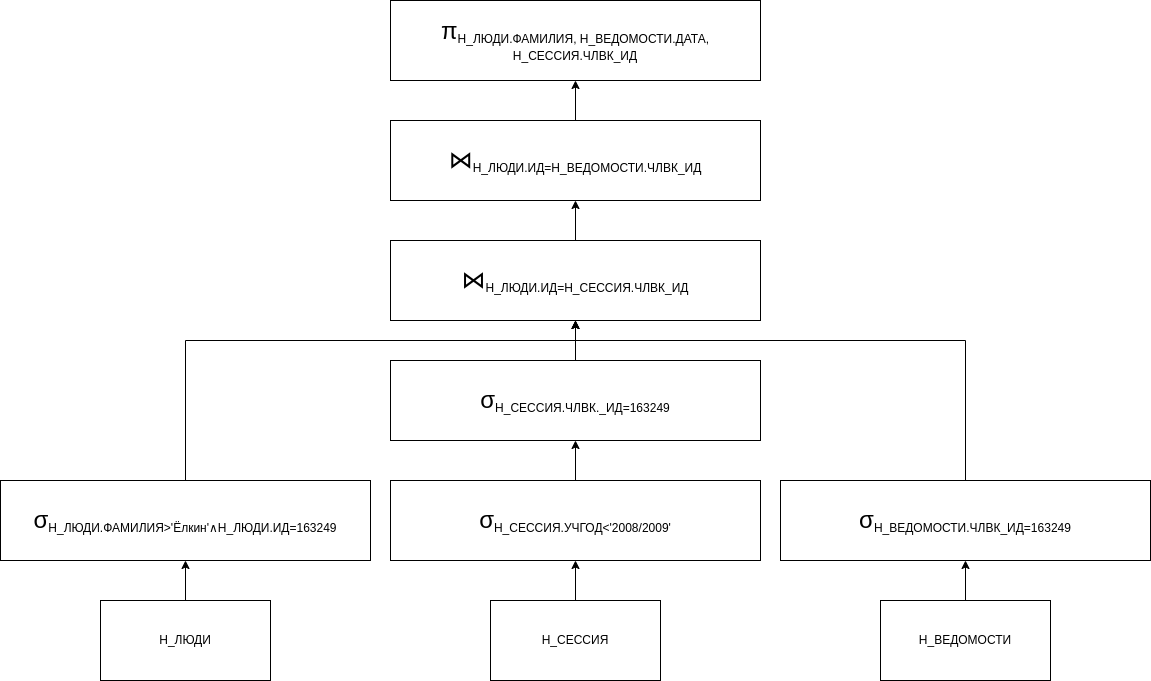


Рисунок 7

# Анализ выбора планировщика

Далее будет представлен результат вывода команды EXPLAIN ANALYZE для исходных запросов.

## Запрос 1

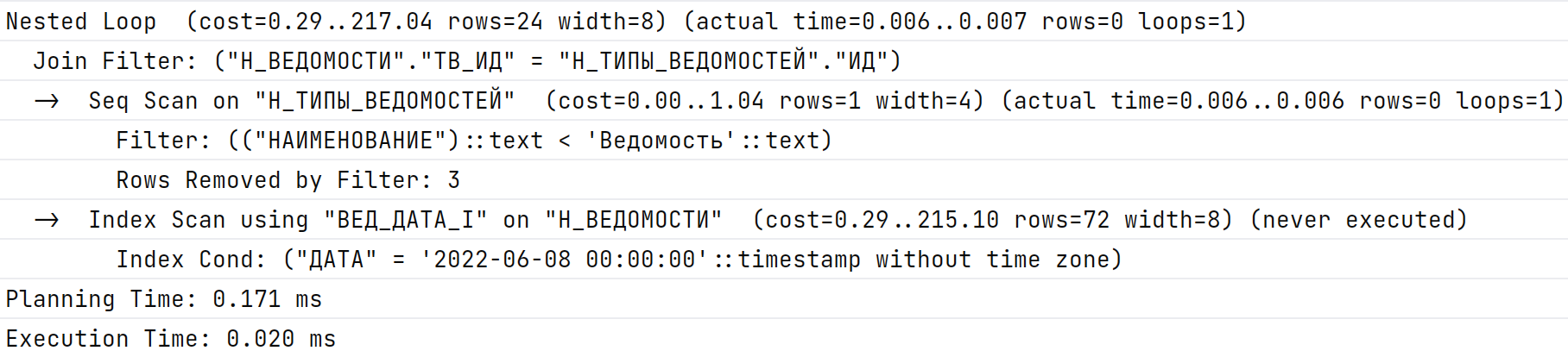


Рисунок 8

## Запрос 2

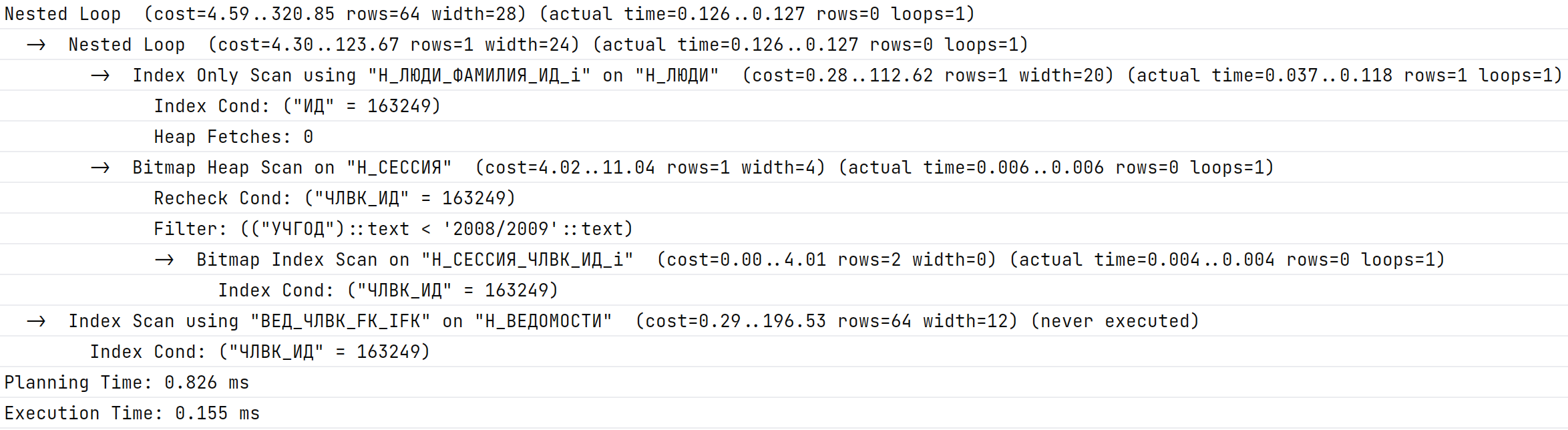


Рисунок 9

# **Заключение**

В результате выполнения данной лабораторной работы изучена теория оптимизации реляционных баз данных. Получены знания о планировании запросов, а также на примере реальных SQL-запросов проработаны возможные для них планы и выбраны эффективные. Главным объектом изучения стали индексы. Рассмотрены основные алгоритмы, на которых они могут быть построены, рамки их применения, дополнительные свойства. Индексы были применены к атрибутам некоторых таблиц БД, в результате чего исходные запросы начали выполняться быстрее.