Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.04 Программная инженерия

**Отчет**

**по лабораторной работе №4**

**по дисциплине «Базы данных»**

Вариант: 5343

Выполнила

Махнева Ирина Александровна,

Группа Р3117

Преподаватель

Чупанов Аликылыч Алибекович

г. Санкт-Петербург, 2025г

Оглавление

[Текст задания 3](#_Toc198314158)

[Реализация запросов на SQL 4](#_Toc198314159)

[Уменьшение времени выполнения запроса 5](#_Toc198314160)

[Первый запрос 5](#_Toc198314161)

[Индексы, добавление которых уменьшит время выполнения запроса: 5](#_Toc198314162)

[Возможные планы выполнения запросов при отсутствии индексов 5](#_Toc198314163)

[План выполнения запроса 6](#_Toc198314164)

[Второй запрос 7](#_Toc198314165)

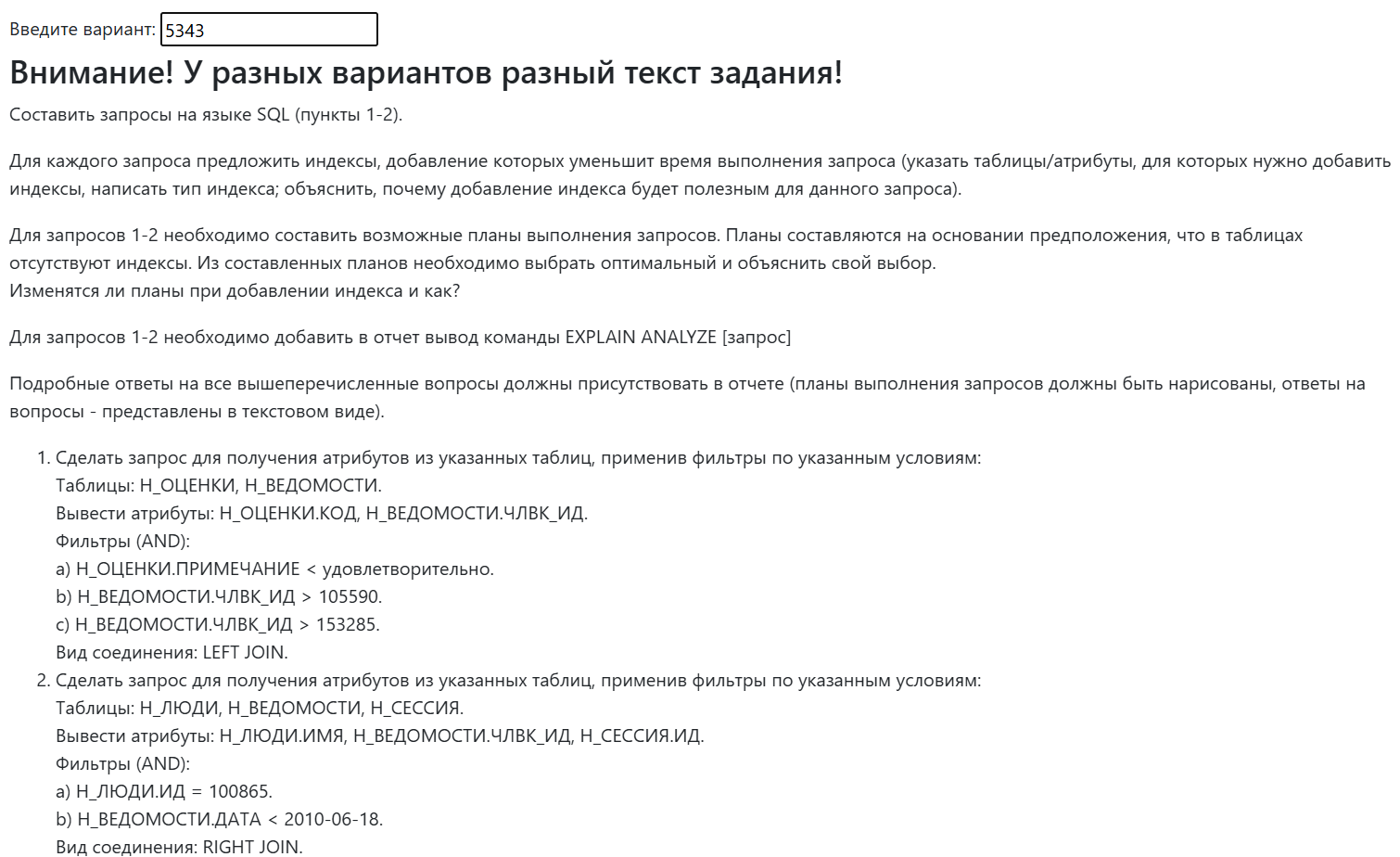
[Индексы, добавление которых уменьшит время выполнения запроса: 7](#_Toc198314166)

[Возможные планы выполнения запросов при отсутствии индексов 8](#_Toc198314167)

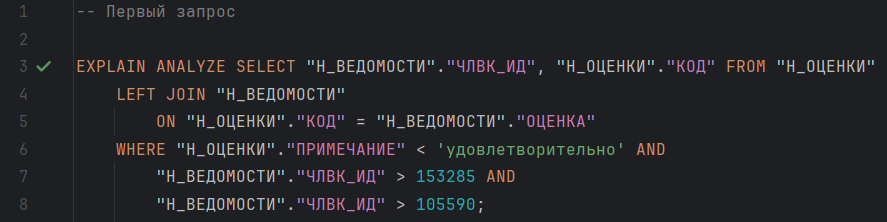
[План выполнения запроса 9](#_Toc198314168)

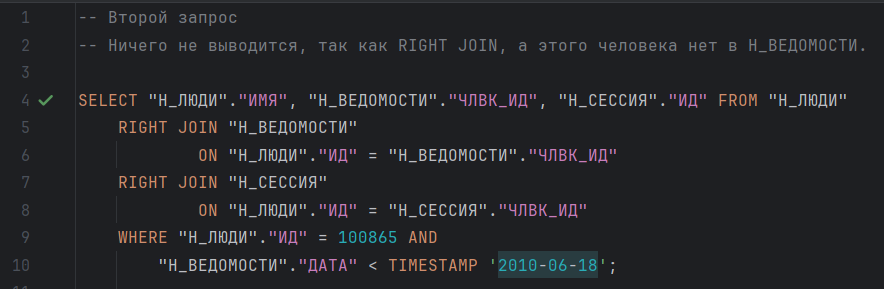
[Вывод 11](#_Toc198314169)

# Текст задания



# Реализация запросов на SQL





# Уменьшение времени выполнения запроса

## Первый запрос

### Индексы, добавление которых уменьшит время выполнения запроса:

1. “Н\_ОЦЕНКИ”
   1. Добавим индекс для атрибутов (ПРИМЕЧАНИЕ, КОД)  
      Тип индекса – B-tree  
      Первый атрибут в индексе – ПРИМЕЧАНИЕ – нужен чтобы в фильтре "Н\_ОЦЕНКИ"."ПРИМЕЧАНИЕ" < ‘удовлетворительно' быстро отобрались сразу только необходимые нам строки.   
      Второй атрибут в индексе – КОД – необходим для того, чтобы при join не обращаться к самой таблице. План сможет выполнить запрос, используя только данные из индекса (Index Only Scan).
2. “Н\_ВЕДОМОСТИ”
   1. Добавим индекс для атрибутов (ЧЛВК\_ИД, ОЦЕНКА)  
      Тип индекса – B-tree.  
      Первый атрибут в индексе – ЧЛВК\_ИД – позволит плану сразу отобрать только необходимые строки.  
      Второй атрибут в индексе – ОЦЕНКА – необходим, чтобы выполнить Index Only Scan и не обращаться к таблице.

Добавление данных индексов типа B-tree позволит уменьшить количество просматриваемых страниц, ускорит фильтрацию данных и выполнение join в запросе.

### Возможные планы выполнения запросов при отсутствии индексов

План 1

1. Скан таблицы “Н\_ОЦЕНКИ”.
2. Скан таблицы “Н\_ВЕДОМОСТИ”.
3. LEFT JOIN (Nested Loop Join) по атрибутам: "Н\_ОЦЕНКИ"."КОД" = "Н\_ВЕДОМОСТИ"."ОЦЕНКА".
4. Фильтрация "Н\_ОЦЕНКИ"."ПРИМЕЧАНИЕ" < 'удовлетворительно'.
5. Фильтрация "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 105590.
6. Фильтрация "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 153285.

План 2

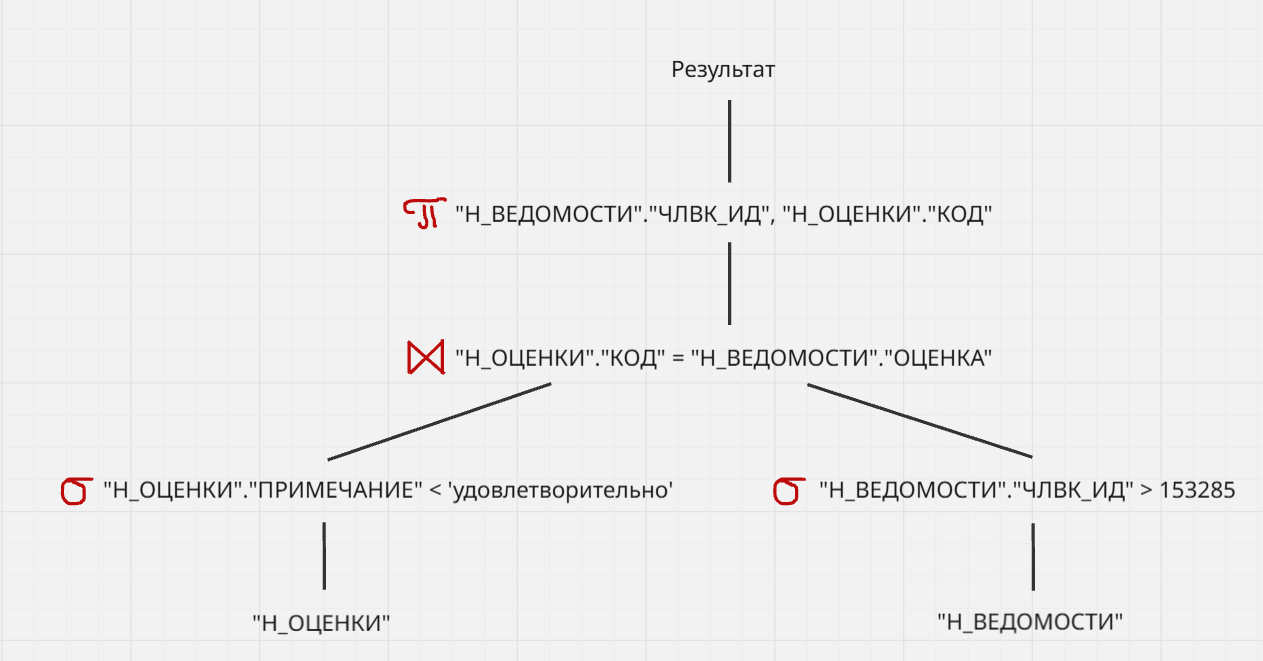
1. Скан таблицы “Н\_ОЦЕНКИ” и фильтрация "Н\_ОЦЕНКИ"."ПРИМЕЧАНИЕ" < 'удовлетворительно'
2. Скан таблицы “Н\_ВЕДОМОСТИ” и фильтрация "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД" > 153285.
3. LEFT JOIN (Nested Loop) по атрибутам: "Н\_ОЦЕНКИ"."КОД" = "Н\_ВЕДОМОСТИ"."ОЦЕНКА".

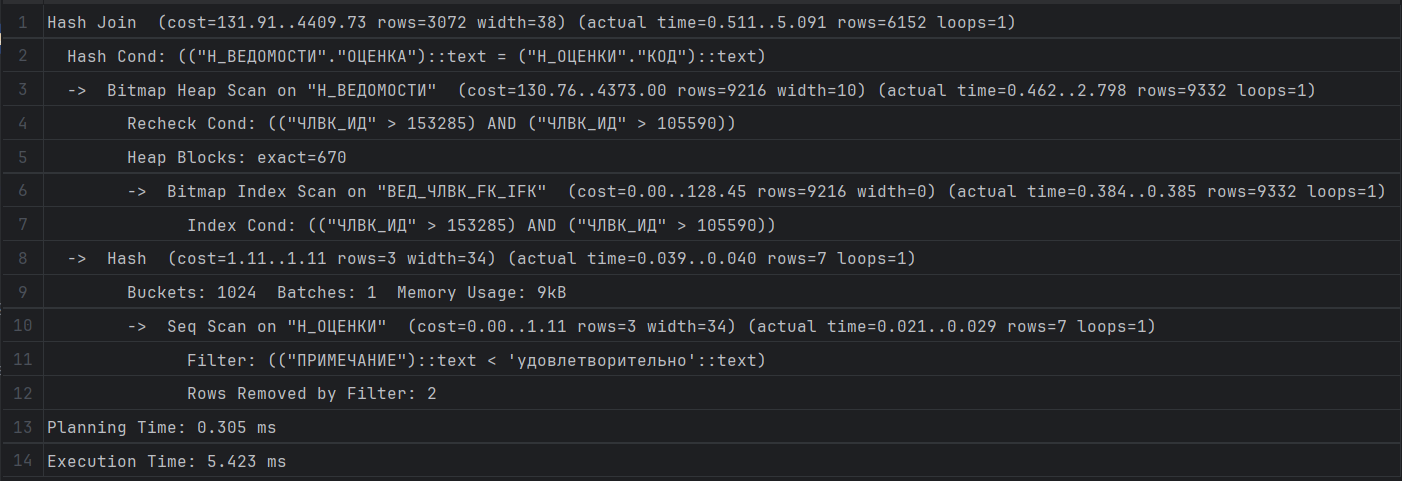
Оптимальный план – план 2, так как происходит фильтрация обеих таблиц до их соединения, для того чтобы в соединении участвовало гораздо меньше строк, уменьшилось количество сравнений и количество промежуточных данных. Операция соединения гораздо более затратная, чем операция фильтрации, поэтому будет выгоднее сначала отфильтровать данные, а уже потом делать join таблиц.

При добавлении индексов планы выполнения запросов изменятся:

1. Вместо полного сканирования “Н\_ОЦЕНКИ” выполняется Index Only Scan, который позволяет взять сразу только те коды, у которых выполнено условие на “ПРИМЕЧАНИЕ” ("ПРИМЕЧАНИЕ" < 'удовлетворительно'). Фильтрация ускорится.
2. Ускорится соединение таблиц благодаря тому, что атрибут “КОД” присутствует в индексе и обращение к таблице не требуется, так как все данные находятся в индексе.
3. Аналогично ускорится работа с таблицей “Н\_ВЕДОМОСТИ”

### План выполнения запроса





## Второй запрос

### Индексы, добавление которых уменьшит время выполнения запроса:

1. “Н\_ЛЮДИ”
   1. Добавим индекс для атрибутов (ИД, ИМЯ)  
      Тип индекса – Hash   
      Первый атрибут в индексе – ИД – нужен чтобы в фильтре "Н\_ЛЮДИ"."ИД" = 100865 быстро отобралась только необходимая строка.   
      Второй атрибут в индексе – ИМЯ – выводится в SELECT, поэтому необходим для того, чтобы не обращаться к самой таблице и использовать данные только из индекса (Index Only Scan).
2. “Н\_ВЕДОМОСТИ”
   1. Добавим индекс для атрибутов (ДАТА, ЧЛВК\_ИД)  
      Тип индекса – B-tree.  
      Первый атрибут в индексе – ДАТА – позволит быстро отфильтровать данные с помощью фильтра "Н\_ВЕДОМОСТИ"."ДАТА" < TIMESTAMP '2010-06-18'.  
      Второй атрибут в индексе - ЧЛВК\_ИД – необходим для того, чтобы при join не обращаться к самой таблице. План сможет выполнить запрос, используя только данные из индекса (Index Only Scan).
3. “Н\_СЕССИЯ”
   1. Добавим индекс для атрибутов (ЧЛВК\_ИД, ИД)  
      Тип индекса – B-tree  
      Первый атрибут в индексе – ЧЛВК\_ИД – необходим для того, чтобы при join не обращаться к самой таблице. План сможет выполнить запрос, используя только данные из индекса (Index Only Scan).  
      Второй атрибут в индексе – ИД – выводится в SELECT, поэтому необходим для Index Only Scan.

### Возможные планы выполнения запросов при отсутствии индексов

План 1:

1. Скан таблицы “Н\_ЛЮДИ”
2. Скан таблицы “Н\_ВЕДОМОСТИ”
3. RIGHT JOIN (Nested Loop) по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"
4. Скан таблицы “Н\_СЕССИЯ”
5. RIGHT JOIN (Nested Loop) по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД"
6. Фильтрация "Н\_ЛЮДИ"."ИД" = 100865
7. Фильтрация "Н\_ВЕДОМОСТИ"."ДАТА" < TIMESTAMP '2010-06-18'

План 2:

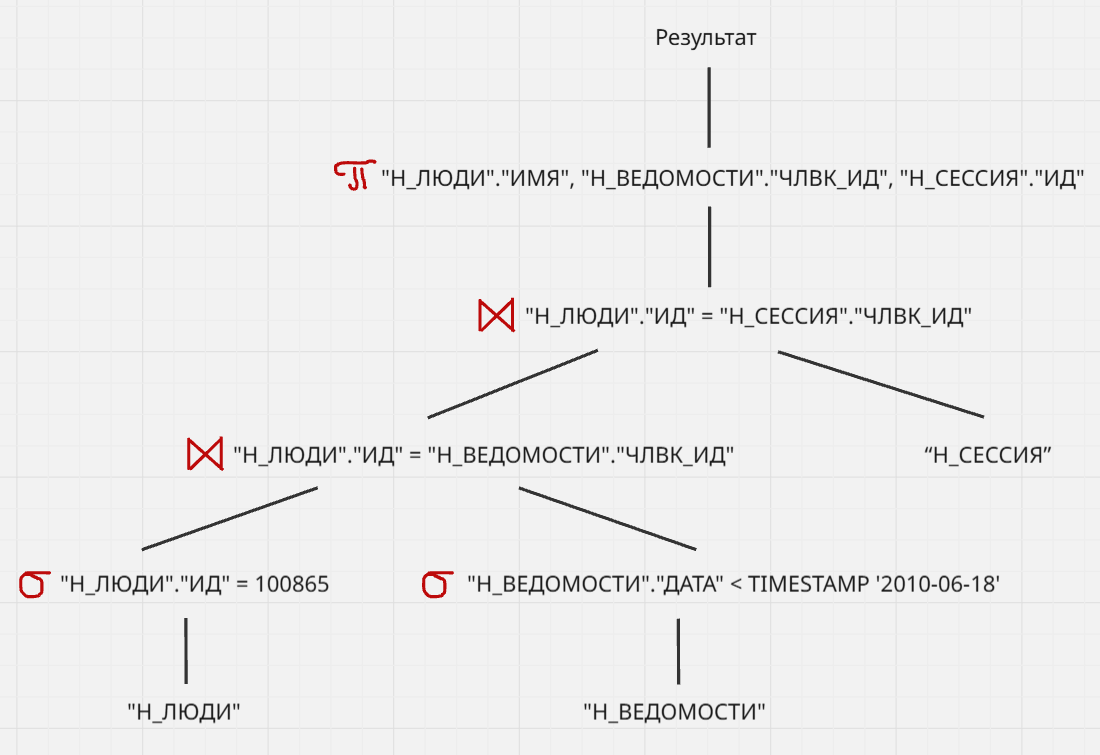
1. Скан таблицы “Н\_ЛЮДИ” и фильтрация "Н\_ЛЮДИ"."ИД" = 100865
2. Скан таблицы “Н\_ВЕДОМОСТИ” и фильтрация "Н\_ВЕДОМОСТИ"."ДАТА" < TIMESTAMP '2010-06-18'
3. RIGHT JOIN (Nested Loop Join) по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД"
4. Скан таблицы “Н\_СЕССИЯ”
5. RIGHT JOIN (Nested Loop Join) по атрибутам "Н\_ЛЮДИ"."ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД"

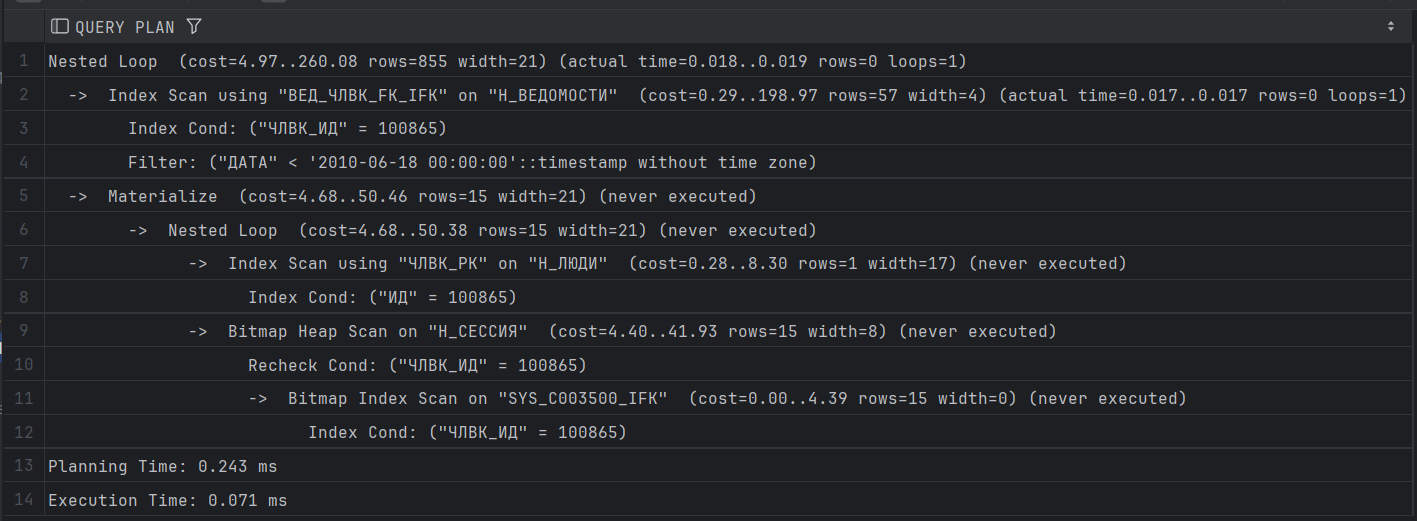
Оптимальный план – план 2, так как происходит фильтрация таблиц до их соединения, для того чтобы в соединении участвовало гораздо меньше строк, уменьшилось количество сравнений и количество промежуточных данных. Операция соединения гораздо более затратная, чем операция фильтрации, поэтому будет выгоднее сначала отфильтровать данные, а уже потом делать join таблиц.

При добавлении индексов планы выполнения запросов изменятся:

1. С помощью индексов будет быстрее произведена фильтрация по атрибуту "Н\_ЛЮДИ"."ИД
2. Также будет ускорена фильтрация по "Н\_ВЕДОМОСТИ"."ДАТА"
3. Ускорится соединение таблиц благодаря тому, что атрибуты присутствует в индексах и обращение к таблице не требуется.

### План выполнения запроса





# Вывод

В ходе выполнения лабораторной работы я изучила различные виды индексов и использовала их для оптимизации скорости выполнения запросов, научилась строить планы выполнения запросов и освоила работу с реляционной алгеброй.