Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №4132

Лабораторная работа №4

По дисциплине

Базы Данных

Выполнил студент группы P3117:

Изаак Герман Константинович

Преподаватель:

Чупанов Аликылыч Алибекович

Санкт-Петербург 2025 г.

1. **Текст задания**

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:  
   Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ОЦЕНКИ.ПРИМЕЧАНИЕ = отлично.  
   b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 105590.  
   c) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 163249.  
   Вид соединения: RIGHT JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
   Вывести атрибуты: Н\_ЛЮДИ.ИМЯ, Н\_ОБУЧЕНИЯ.НЗК, Н\_УЧЕНИКИ.НАЧАЛО.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО < Георгиевич.  
   b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 105590.  
   Вид соединения: LEFT JOIN.
3. **Реализация запросов на SQL**

Изображение выглядит как текст, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Первый запрос**

Индексы, добавление которых уменьшит время выполнения запроса:

«Н\_ОЦЕНКИ»:

* Индекс на атрибутах **(ПРИМЕЧАНИЕ, КОД)** *(Hash)*: первый столбец — фильтр ПРИМЕЧАНИЕ = 'отлично', поэтому план сможет сразу отобрать только нужные строки. Второй столбец КОД входит в условие соединения, что позволит выполнить «Index Only Scan» без обращения к таблице.

«Н\_ВЕДОМОСТИ»:

* Индекс на атрибутах **(ЧЛВК\_ИД, ОЦЕНКА)** *(B-tree)*: ведущий столбец ускоряет диапазонный фильтр ЧЛВК\_ИД < 105 590, а включённый ОЦЕНКА совпадает с полем JOIN.

Добавление указанных B-tree-индексов снизит объём просматриваемых страниц и ускорит фильтрацию и соединение в запросе.

Возможные планы выполнения запросов без индексов:

План 1:

* Полный скан таблицы **Н\_ОЦЕНКИ** (без фильтра).
* Полный скан таблицы **Н\_ВЕДОМОСТИ** с применением фильтра по условию ЧЛВК\_ИД < 105 590
* RIGHT JOIN по полю КОД = ОЦЕНКА
* После соединения выполняется фильтрация результата: ПРИМЕЧАНИЕ = 'отлично'

План 2:

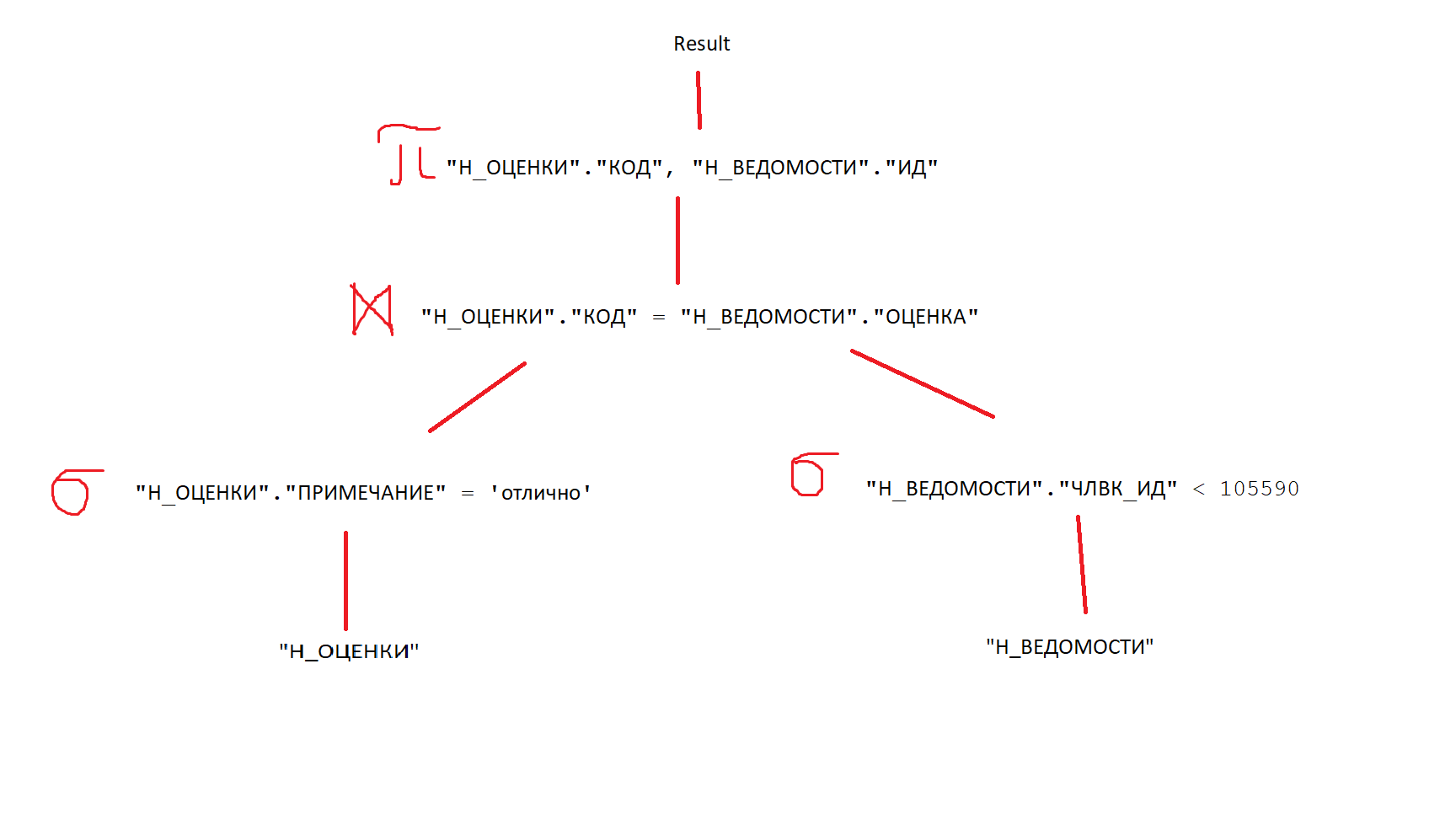
* Полный скан **Н\_ОЦЕНКИ** + сразу фильтр ПРИМЕЧАНИЕ = 'отлично'
* Полный скан **Н\_ВЕДОМОСТИ** + сразу фильтр ЧЛВК\_ИД < 105 590.
* RIGHT JOIN по КОД = ОЦЕНКА уже над уменьшенными наборами строк.

Оптимальный план — План 2. Фильтрация выполняется до соединения, поэтому в JOIN участвует гораздо меньше строк; уменьшаются сравнения и объём промежуточных данных.

При добавлении индексов планы выполнения запросов изменятся:

* Вместо полного сканирования «Н\_ОЦЕНКИ» выполняется **Index Only Scan** — сразу берутся только коды с пометкой *«отлично»*.
* Соединение по-прежнему реализуется **Nested Loop Join**, но теперь каждая итерация обращается к точному диапазону в индексе, а не просматривает всю таблицу.
* За счёт индексных обращений объём читаемых страниц резко уменьшается, и время выполнения запроса снижается в разы.

План выполнения запроса



Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Второй запрос**

Индексы, добавление которых уменьшит время выполнения запроса:

**«**Н\_ЛЮДИ**»**

* Индекс на атрибутах (**ОТЧЕСТВО, ИД**) (*B-tree*): Фильтр ОТЧЕСТВО < 'Георгиевич' использует ведущее поле; ИД сразу доступен для присоединения к «Н\_ОБУЧЕНИЯ», что уменьшит число обращений к таблице «Н\_ЛЮДИ».

«Н\_ОБУЧЕНИЯ»

* Индекс на атрибутах (**ЧЛВК\_ИД, ВИД\_ОБУЧ\_ИД, НЗК**) (*B-tree*): Первые два столбца совпадают с парой, по которой Н\_УЧЕНИКИ присоединяется к Н\_ОБУЧЕНИЯ, значит, SQL сможет взять строку «обучения» по точному ключу. Третий столбец (НЗК) выводится в SELECT, поэтому планер сможет выполнить Index-Only Scan без обращения к таблице.

«Н\_УЧЕНИКИ»

* Индекс на атрибутах (**ЧЛВК\_ИД, ВИД\_ОБУЧ\_ИД, НАЧАЛО**) (*B-tree*): Первые два столбца — точный ключ соединения с Н\_ОБУЧЕНИЯ. В индекс включён столбец НАЧАЛО, который нужен в выходном наборе, поэтому чтение тоже может быть «index-only».

Таблицы больше не читаются полностью: каждая операция соединения задействует точечные обращения по составному ключу, а выборка нужных полей выполняется прямо из индексных страниц.

Возможные планы выполнения запросов без индексов:

План 1:

* Полный скан таблицы «Н\_ЛЮДИ» без фильтров
* Левое соединение (**LEFT JOIN**) с **Н\_ОБУЧЕНИЯ** — полный скан, соединение по ИД = ЧЛВК\_ИД
* Левое соединение с **Н\_УЧЕНИКИ** — полный скан, соединение по паре (ЧЛВК\_ИД,ВИД\_ОБУЧ\_ИД).
* После всех соединений выполняется фильтрация результата:  
   "ОТЧЕСТВО" < 'Георгиевич' и "ЧЛВК\_ИД" > 105 590.

План 2:

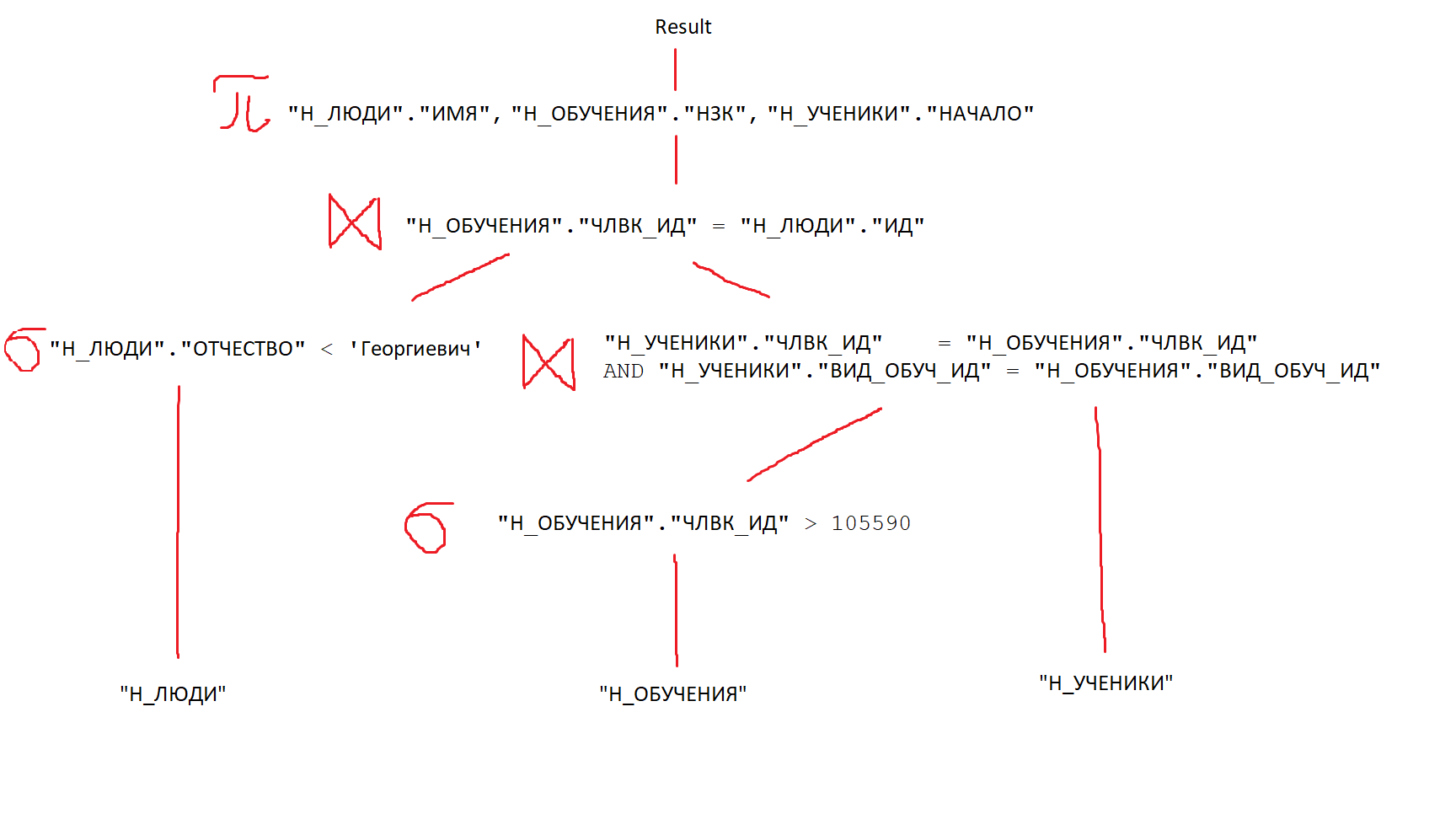
* Полный скан таблицы «**Н\_ЛЮДИ**» и применение фильтра ОТЧЕСТВО < 'Георгиевич'
* Полный скан таблицы «**Н\_ОБУЧЕНИЯ**» и применения фильтра ЧЛВК\_ИД > 105 590
* LEFT JOIN (Nested Loop) «отфильтрованных» **Н\_ЛЮДИ ⟶ Н\_ОБУЧЕНИЯ**.
* LEFT JOIN (Nested Loop) результата с отфильтрованной подмножеством Н\_УЧЕНИКИ

Оптимальный план — План 2. Фильтрация выполняется до соединений, поэтому каждая последующая операция обрабатывает существенно меньше строк; объём промежуточных данных и число сравнений минимальны.

При добавлении индексов планы выполнения запросов изменятся:

* Соединения выполняются тем же **Nested Loop**, но каждая итерация обращается к строкам «Н\_ОБУЧЕНИЯ» и «Н\_УЧЕНИКИ» по точному ключу, без повторного прохода по таблице.
* Промежуточные наборы резко сокращаются; общее время запроса уменьшается

План выполнения запроса



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Вывод по работе**

В ходе выполнения лабораторной работы я освоил работу с реляционной алгеброй и научился строить планы выполнения запросов, а также их диаграммы. Я изучил различные виды индексов и узнал, как использовать их для оптимизации скорости выполнения запросов. Теперь я могу применять полученные знания для эффективной работы с базами данных и повышения производительности SQL-запросов.