

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Санкт-Петербургский национальный исследовательский университет  
ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

## **ЛАБОРАТОРНАЯ РАБОТА № 4**

по дисциплине  
‘Базы данных’

Вариант №3542

*Выполнил:*

Студент группы Р3107  
Садовой Григорий  
Владимирович

*Преподаватель:*

Бострикова Д. К.



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург, 2023

## 1. Текст задания

Для выполнения лабораторной работы №4 необходимо:

- 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.

## 2. Реализация запросов на SQL

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

Таблицы: Н\_ЛЮДИ, Н\_СЕССИЯ.

Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_СЕССИЯ.ДАТА.

Фильтры (AND):

а) Н\_ЛЮДИ.ИД = 100012.

б) Н\_СЕССИЯ.УЧГОД > 2001/2002.

Вид соединения: RIGHT JOIN.

```
SELECT
  Н_ЛЮДИ.ОТЧЕСТВО,
  Н_СЕССИЯ.ДАТА
FROM
  Н_ЛЮДИ
RIGHT JOIN
  Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД
WHERE
  Н_ЛЮДИ.ИД = 100012 AND
  Н_СЕССИЯ.УЧГОД > '2001/2002';
```

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.

Вывести атрибуты: Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

а) Н\_ЛЮДИ.ИМЯ > Владимир.

б) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 112514.

с) Н\_УЧЕНИКИ.ИД = 250098.

Вид соединения: INNER JOIN.

```
SELECT Н_ЛЮДИ.ФАМИЛИЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.НАЧАЛО
FROM Н_ЛЮДИ
INNER JOIN Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
INNER JOIN Н_УЧЕНИКИ ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ИМЯ > 'Владимир' AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 112514
AND Н_УЧЕНИКИ.ИД = 250098;
```

### 3. Уменьшение выполнения времени 1 запроса

Индексы, добавление которых уменьшит время выполнения запроса:

а) На таблице Н\_ЛЮДИ:

- Индекс на атрибуте ИД (B-tree) Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ИД = 100012 .

б) На таблице Н\_СЕССИЯ:

- Индекс на атрибуте УЧГОД (B-tree) Это ускорит фильтрацию строк с условием "Н\_СЕССИЯ.УЧГОД > '2001/2002'".
- Индекс на атрибуте ЧЛВК\_ИД (B-tree) Это ускорит соединение таблиц по атрибуту ЧЛВК\_ИД.

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

**Возможные планы выполнения запросов без индексов:**

План 1:

- Полный скан таблицы Н\_ЛЮДИ.
- Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию "Н\_СЕССИЯ.УЧГОД > '2001/2002'".
- Соединение таблиц с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.
- Фильтрация результата соединения по условию "Н\_ЛЮДИ.ИД = 100012 .

План 2:

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ИД = 100012.
- Полный скан таблицы Н\_СЕССИЯ с применением фильтра по условию "Н\_СЕССИЯ.УЧГОД > '2001/2002'".
- Соединение таблиц с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.

Оптимальный план:

- План 2, потому что фильтрация данных выполняется до соединения таблиц, что уменьшает количество строк для обработки.

При добавлении индексов планы выполнения запросов изменятся:

- Вместо полного скана таблиц будет использоваться индексный скан.
- Nested Loops Join станет быстрее благодаря индексу на атрибуте ЧЛВК\_ИД.



#### 4. План выполнения 1 запроса

##### 1st QUERY PLAN

Nested Loop (cost=4.61..31.54 rows=6 width=28) (actual time=0.059..0.071 rows=6 loops=1)

-> Index Scan using "ЧЛВК\_ПК" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=24) (actual time=0.024..0.024 rows=1 loops=1)

Index Cond: ("ИД" = 100012)

-> Bitmap Heap Scan on "Н\_СЕССИЯ" (cost=4.33..23.18 rows=6 width=12) (actual time=0.030..0.039 rows=6 loops=1)

Recheck Cond: ("ЧЛВК\_ИД" = 100012)

Filter: (("УЧГОД")::text > '2001/2002')::text)

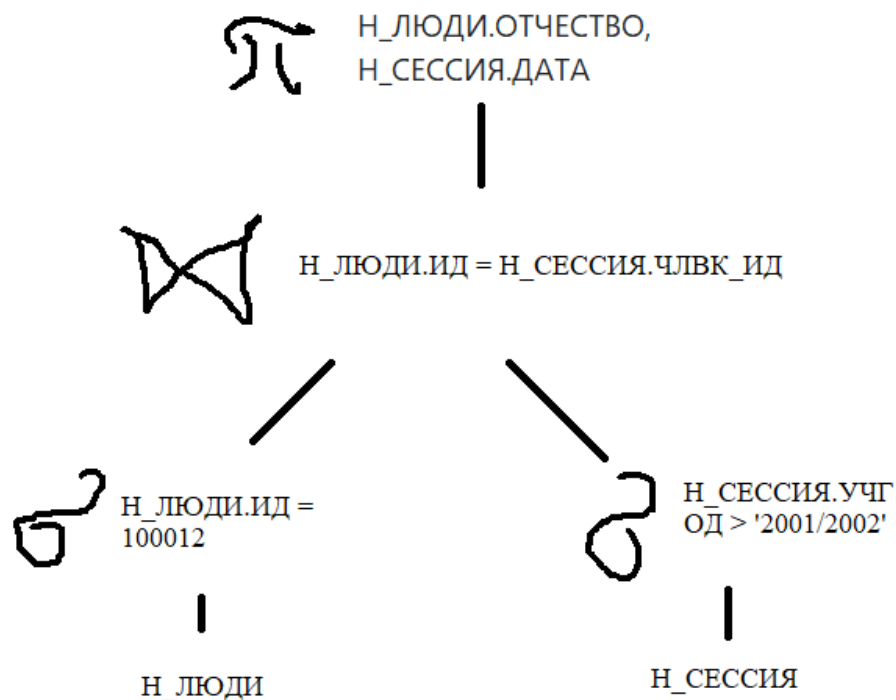
Heap Blocks: exact=5

-> Bitmap Index Scan on "SYS\_C003500\_IFK" (cost=0.00..4.33 rows=6 width=0) (actual time=0.019..0.019 rows=6 loops=1)

Index Cond: ("ЧЛВК\_ИД" = 100012)

Planning Time: 0.233 ms

Execution Time: 0.121 ms



## 5. Уменьшение выполнения времени 2 запроса

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

а) На таблице Н\_ЛЮДИ:

- Индекс на атрибуте ИМЯ (B-tree). Это ускорит фильтрацию строк с условием "Н\_ЛЮДИ.ИМЯ > 'Владимир'".

б) На таблице Н\_ОБУЧЕНИЕ:

- Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_ОБУЧЕНИЕ.ЧЛВК\_ИД > 112534".

в) На таблице Н\_УЧЕНИКИ:

- Индекс на атрибуте ЧЛВК\_ИД (B-tree). Это ускорит фильтрацию строк с условием "Н\_УЧЕНИКИ.ИД = 250098".

Добавление индексов на указанные столбцы позволит ускорить поиск нужных записей по фильтрам в запросе.

### Возможные планы выполнения запросов без индексов:

План 1:

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ИМЯ > 'ВЛАДИМИР'".
- Полный скан таблицы Н\_ОБУЧЕНИЕ с применением фильтра по условию "Н\_ОБУЧЕНИЕ.ЧЛВК\_ИД > 112534".
- Соединение таблиц Н\_ЛЮДИ и ОБУЧЕНИЕ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.
- Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "Н\_УЧЕНИКИ.ИД = 250098".
- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНЕКИ с использованием Nested Loops Join по атрибуту ЧЛВК\_ИД.

План 2:

- Полный скан таблицы Н\_ЛЮДИ с применением фильтра по условию "Н\_ЛЮДИ.ИМЯ > 'Владимир'".
- Полный скан таблицы Н\_ОБУЧЕНИЕ с применением фильтра по условию "Н\_ОБУЧЕНИЕ.ЧЛВК\_ИД > 112534".
- Соединение таблиц Н\_ЛЮДИ и Н\_ОБУЧЕНИЕ с использованием Hash Join по атрибуту ЧЛВК\_ИД.
- Полный скан таблицы Н\_УЧЕНИКИ с применением фильтра по условию "Н\_УЧЕНИКИ.ИД = 250098".
- Соединение результатов предыдущего этапа с таблицей Н\_УЧЕНИЕИ с использованием Hash Join по атрибуту ЧЛВК\_ИД.

Оптимальный план:

- План 1, так как Nested Loops Join предпочтительнее Hash Join в случаях, когда обрабатываемые наборы данных небольшие, и заранее известно, что результаты фильтрации будут содержать малое количество строк..

При добавлении индексов планы выполнения запросов изменятся:

- Вместо полного скана таблиц будет использоваться индексный скан.
- Nested Loops Join станет быстрее благодаря индексам на атрибутах ЧЛВК\_ИД.

## 6. План выполнения 2 запроса

### 2nd QUERY PLAN

Nested Loop (cost=0.85..13.02 rows=1 width=28) (actual time=0.039..0.040 rows=0 loops=1)

-> Nested Loop (cost=0.57..12.62 rows=1 width=16) (actual time=0.026..0.028 rows=1 loops=1)

-> Index Scan using "УЧЕН\_PK" on "Н\_УЧЕНИКИ" (cost=0.29..8.30 rows=1 width=12) (actual time=0.012..0.013 rows=1 loops=1)

Index Cond: ("ИД" = 250098)

-> Index Only Scan using "ОБУЧ\_ЧЛВК\_FK\_I" on "Н\_ОБУЧЕНИЯ" (cost=0.28..4.30 rows=1 width=4) (actual time=0.007..0.007 rows=1 loops=1)

Index Cond: (("ЧЛВК\_ИД" = "Н\_УЧЕНИКИ"."ЧЛВК\_ИД") AND ("ЧЛВК\_ИД" > 112514))

Heap Fetches: 0

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..0.40 rows=1 width=20) (actual time=0.010..0.010 rows=0 loops=1)

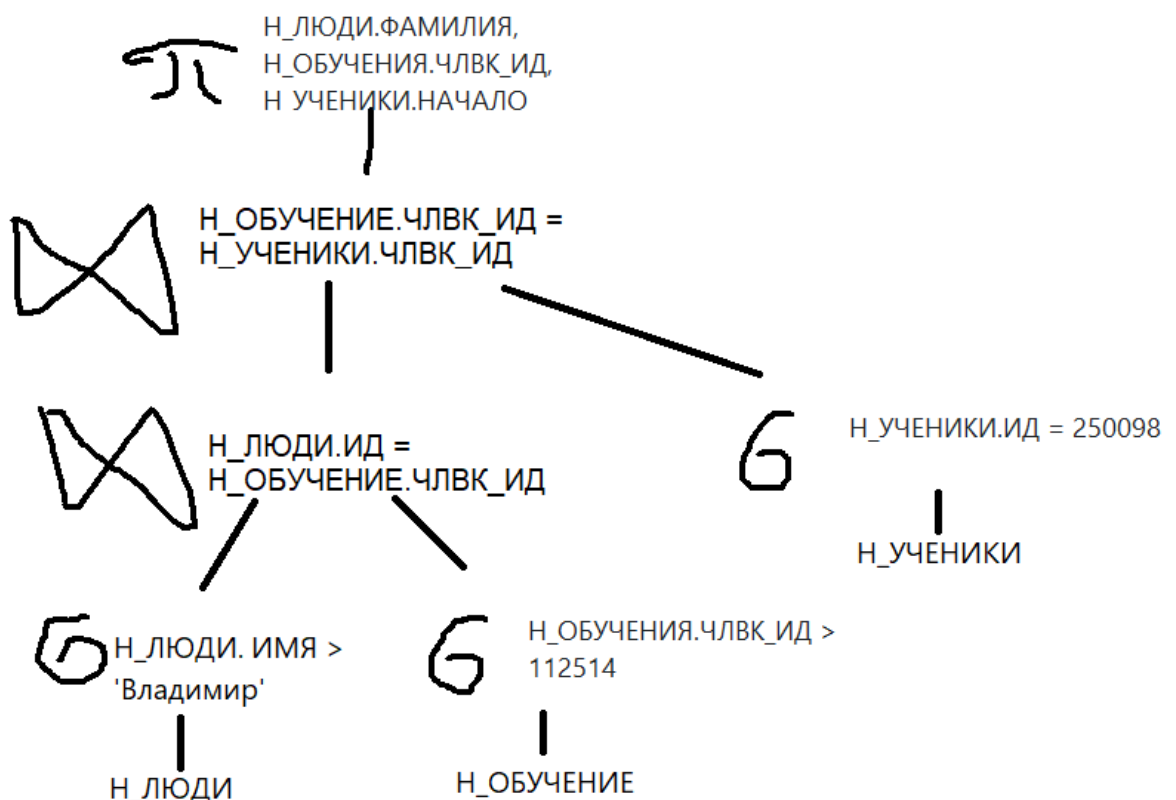
Index Cond: ("ИД" = "Н\_ОБУЧЕНИЯ"."ЧЛВК\_ИД")

Filter: (("ИМЯ")::text > 'Владимир')::text)

Rows Removed by Filter: 1

Planning Time: 0.825 ms

Execution Time: 0.091 ms



## **7. Вывод**

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.



Доп. задание:

|  |
|--|
|  |
|--|

