

**Лабораторная работа № 4**  
по дисциплине базы данных  
Индексы  
Вариант № 1678

Выполнил:  
студент группы Р3116  
Сиразетдинов А. Н  
Преподаватель:  
Горбунов М. В.

# Содержание

<b>1</b>	<b>Текст задания</b>	<b>3</b>
<b>2</b>	<b>Запрос 1</b>	<b>4</b>
2.1	Реализация запроса на SQL . . . . .	4
2.2	Использование индексов . . . . .	4
2.3	Планы выполнения запроса . . . . .	4
2.4	Оптимальный план . . . . .	6
2.5	Выполнение команды EXPLAIN ANALYZE . . . . .	6
<b>3</b>	<b>Запрос 2</b>	<b>7</b>
3.1	Реализация запроса на SQL . . . . .	7
3.2	Использование индексов . . . . .	7
3.3	Планы выполнения запросов . . . . .	8
3.4	Оптимальный план . . . . .	9
3.5	Выполнение команды EXPLAIN ANALYSE . . . . .	9
<b>4</b>	<b>Вывод</b>	<b>10</b>

# 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. Изменяются 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:

Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

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

Фильтры (AND):

(a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > Ведомость.

(b) Н\_ВЕДОМОСТИ.ИД > 1250981.

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.ЧЛВК\_ИД, Н\_УЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

(a) Н\_ЛЮДИ.ФАМИЛИЯ < Ёлкин.

(b) Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД > 112514.

(c) Н\_УЧЕНИКИ.ИД > 100410.

Вид соединения: RIGHT JOIN.

## 2 Запрос 1

### 2.1 Реализация запроса на SQL

```
SELECT
    Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД,
    Н_ВЕДОМОСТИ.ЧЛВК_ИД
FROM
    "Н_ТИПЫ_ВЕДОМОСТЕЙ"
RIGHT JOIN "Н_ВЕДОМОСТИ"
ON
    Н_ВЕДОМОСТИ."ТВ_ИД" = Н_ТИПЫ_ВЕДОМОСТЕЙ."ИД"
WHERE
    Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ > 'Ведомость'
    AND Н_ВЕДОМОСТИ.ИД > 1250981;
```

### 2.2 Использование индексов

#### Таблица Н\_ТИПЫ\_ВЕДОМОСТЕЙ

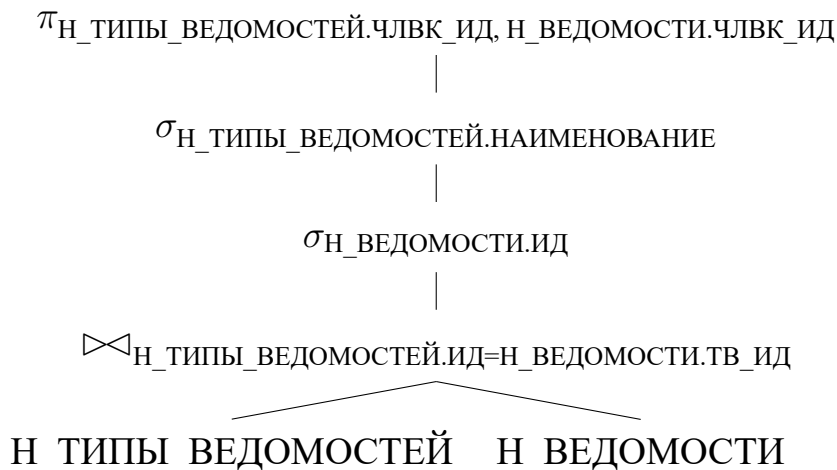
Индекс b-tree на атрибуте НАИМЕНОВАНИЕ уменьшит время выполнения запроса, потому что индекс b-tree хорошо подходит для операций выборки больше определенного значения

#### Таблица Н\_ВЕДОМОСТИ

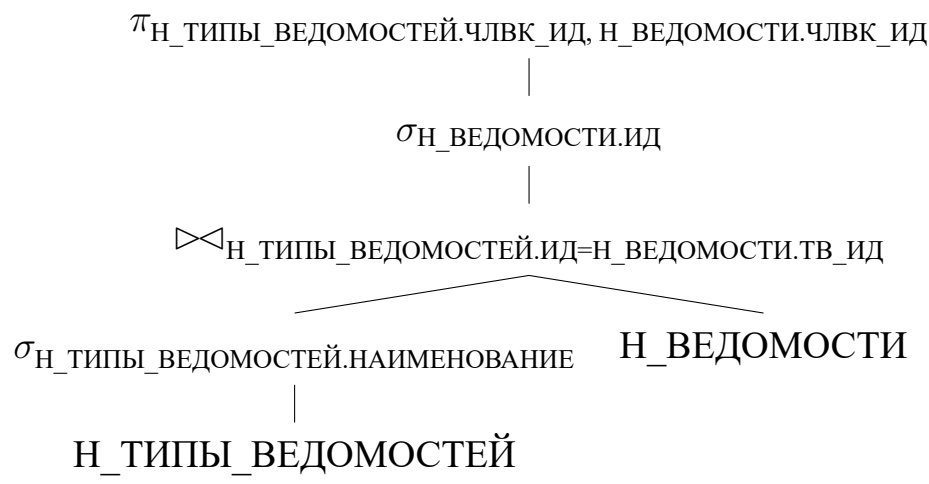
Индекс b-tree на атрибуте ИД уменьшит время выполнения запроса, потому что индекс b-tree хорошо подходит для операций выборки больше определенного значения

### 2.3 Планы выполнения запроса

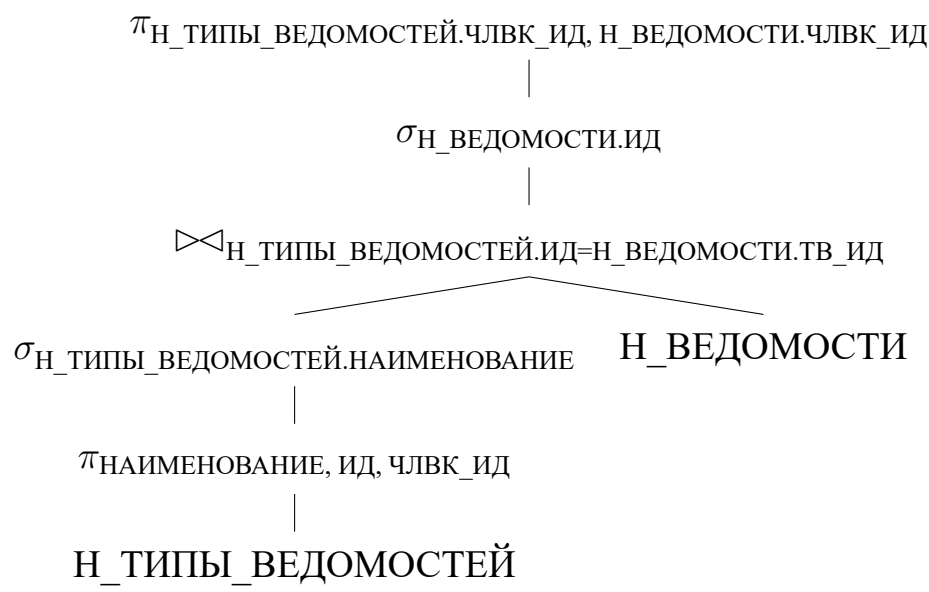
#### План 1



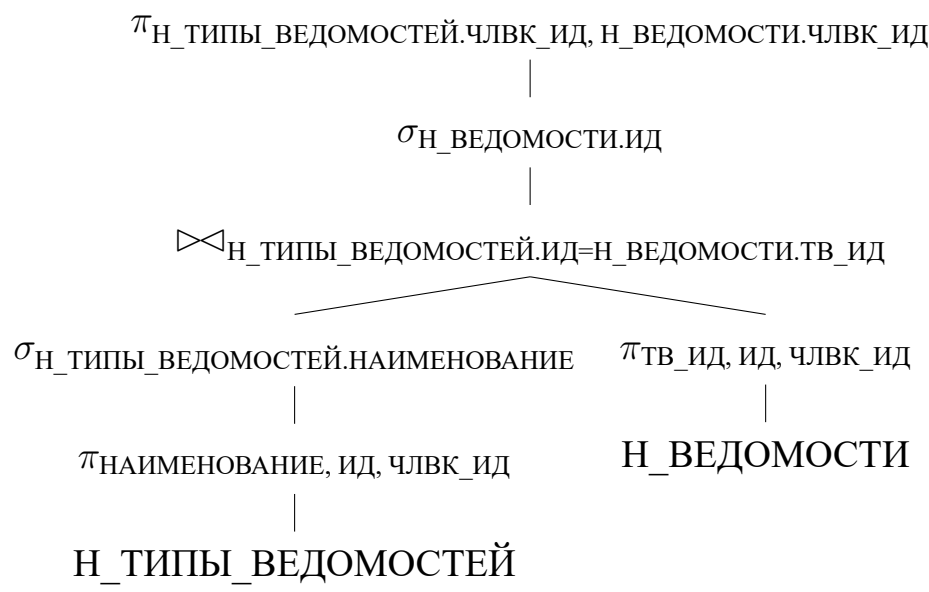
План 2



План 3



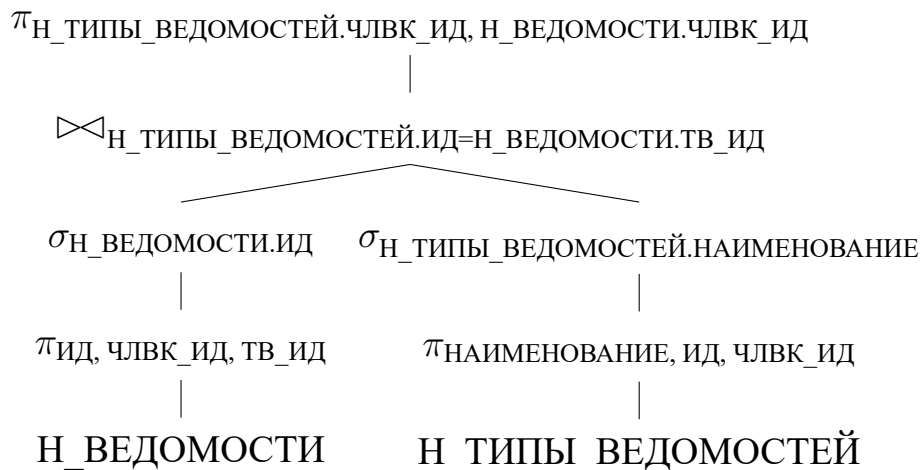
План 4



## План 5



## План 6



## 2.4 Оптимальный план

Самый оптимальный план - План 6, потому что он использует левостороннее дерево и все выборки и проекции сделаны максимально рано.

## 2.5 Выполнение команды EXPLAIN ANALYZE

```
Hash Join (cost=491.19..5047.40 rows=8676 width=8) (actual time=1.650..10.511 rows=6789 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=490.14..4881.51 rows=26029 width=8) (actual time=1.484..5.907 rows=26125 loops=1)
      Recheck Cond: ("ИД" > 1250981)
      Heap Blocks: exact=642
      -> Bitmap Index Scan on "ВЕД_ПК" (cost=0.00..483.64 rows=26029 width=0) (actual time=1.376..1.376 rows=26125 loops=1)
        Index Cond: ("ИД" > 1250981)
    -> Hash (cost=1.04..1.04 rows=1 width=4) (actual time=0.065..0.066 rows=2 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual time=0.053..0.056 rows=2 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text > 'Ведомость'::text)
        Rows Removed by Filter: 1
Planning Time: 1.005 ms
Execution Time: 10.903 ms
```

## 3 Запрос 2

### 3.1 Реализация запроса на SQL

```
SELECT
    Н_ЛЮДИ.ИМЯ,
    Н_ОБУЧЕНИЯ.ЧЛВК_ИД,
    Н_УЧЕНИКИ.НАЧАЛО
FROM
    "Н_ЛЮДИ"
RIGHT JOIN "Н_ОБУЧЕНИЯ"
ON
    Н_ЛЮДИ."ИД" = Н_ОБУЧЕНИЯ."ЧЛВК_ИД"
RIGHT JOIN "Н_УЧЕНИКИ"
ON
    Н_УЧЕНИКИ."ЧЛВК_ИД" = Н_ОБУЧЕНИЯ."ЧЛВК_ИД"
    AND Н_УЧЕНИКИ."ВИД_ОБУЧ_ИД" =
    Н_ОБУЧЕНИЯ."ВИД_ОБУЧ_ИД"
WHERE
    Н_ЛЮДИ.ФАМИЛИЯ < 'Ёлкин'
    AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД > 112514
    AND Н_УЧЕНИКИ.ИД > 100410;
```

### 3.2 Использование индексов

#### Таблица Н\_ЛЮДИ

Индекс b-tree на атрибуте ФАМИЛИЯ уменьшит время выполнения запроса, потому что индекс b-tree хорошо подходит для операций выборки меньше определенного значения

#### Таблица Н\_ОБУЧЕНИЯ

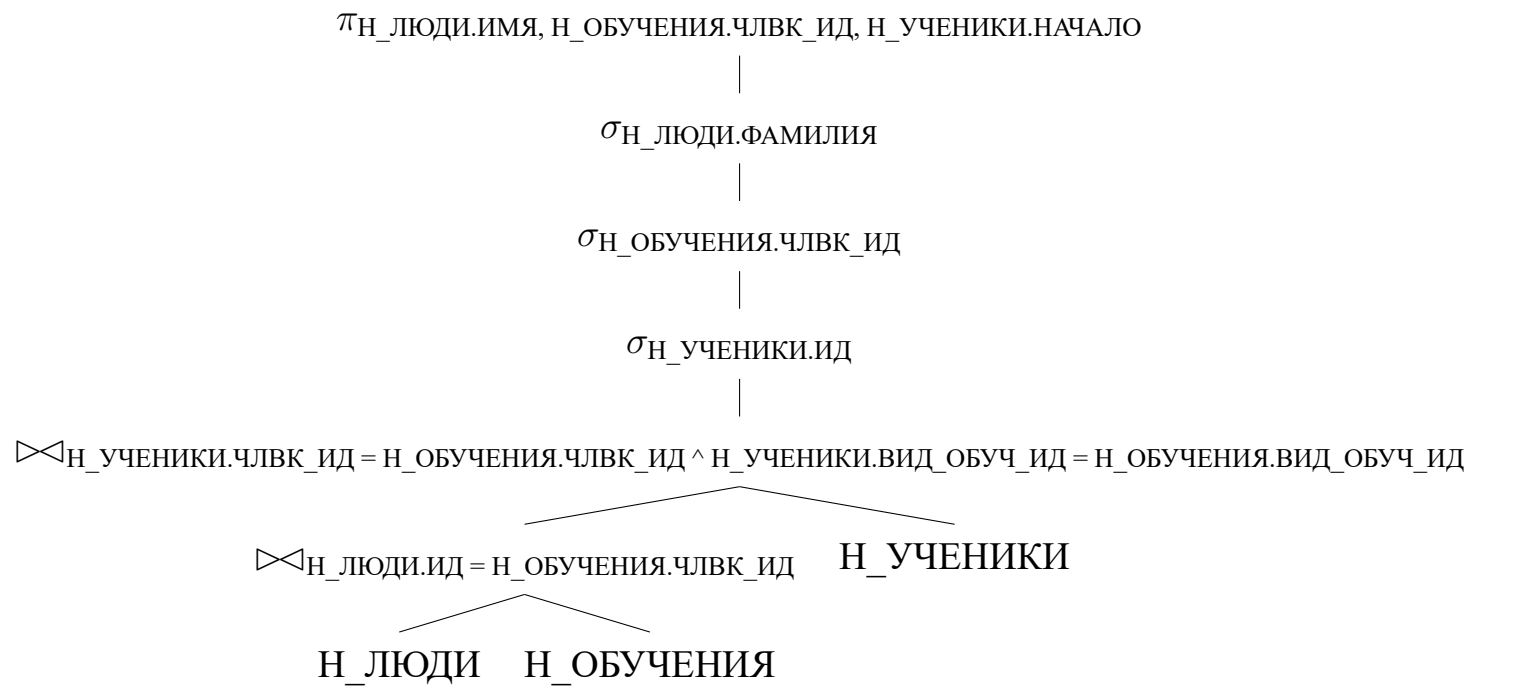
Индекс b-tree на атрибуте ЧЛВК\_ИД уменьшит время выполнения запроса, потому что индекс b-tree хорошо подходит для операций выборки больше определенного значения

#### Таблица Н\_УЧЕНИКИ

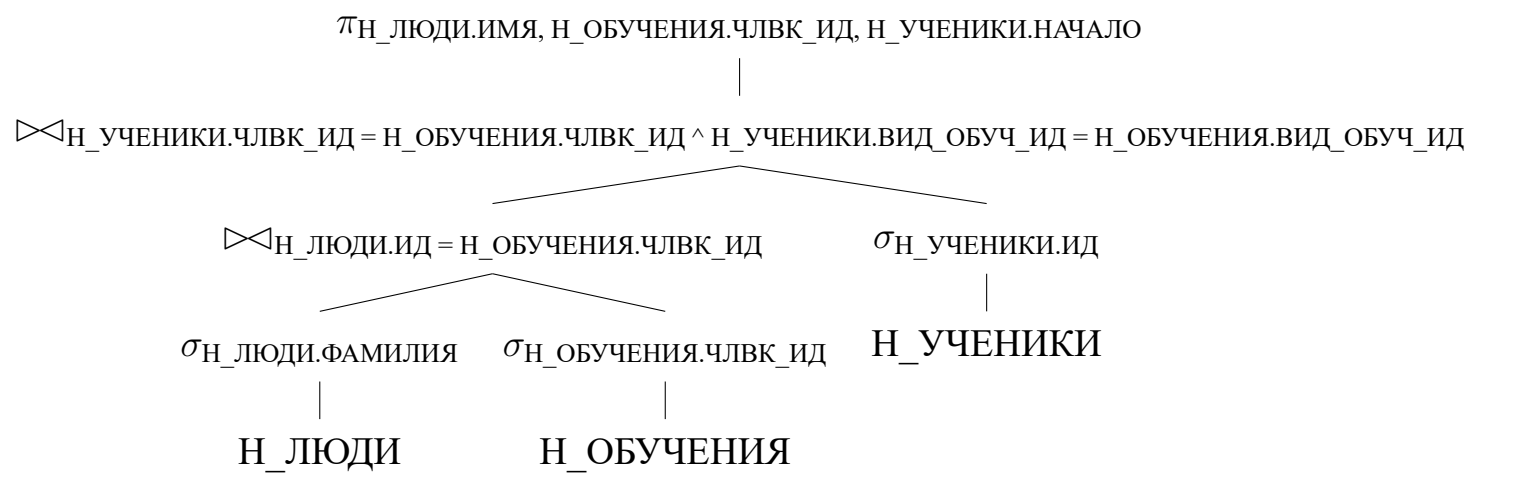
Индекс b-tree на атрибуте ИД уменьшит время выполнения запроса, потому что индекс b-tree хорошо подходит для операций выборки больше определенного значения

### 3.3 Планы выполнения запросов

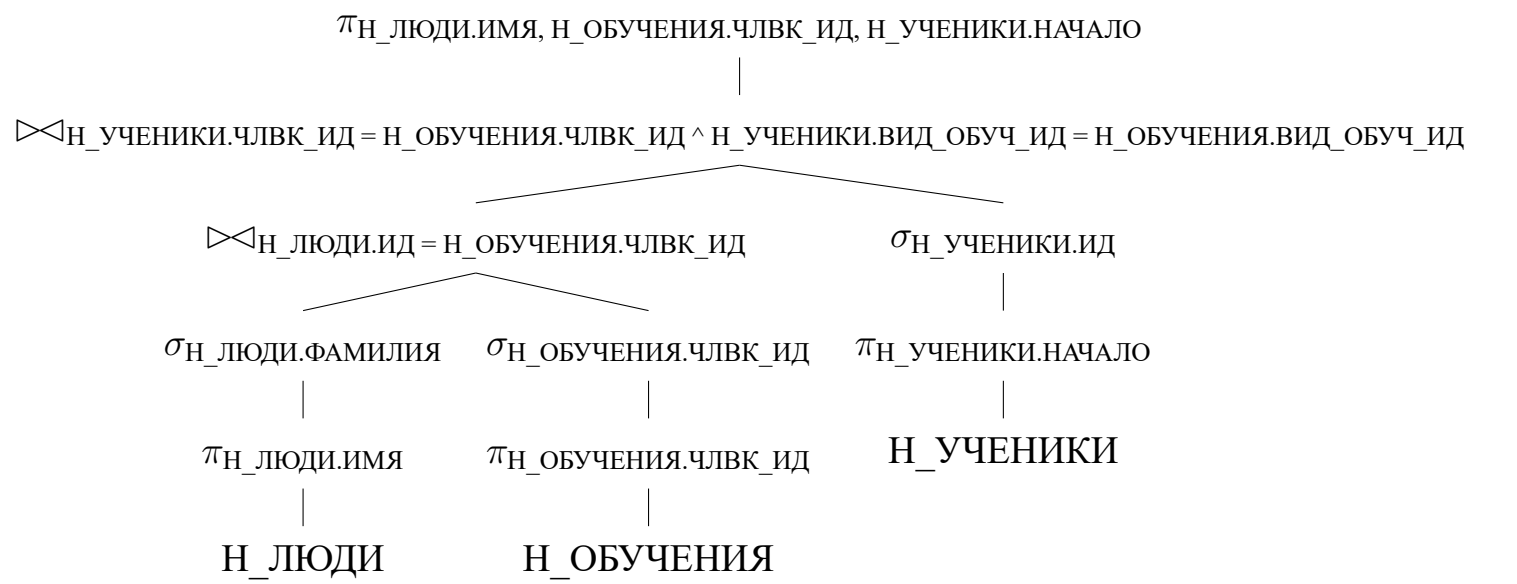
План 1



План 2



План 3





### 3.4 Оптимальный план

Самый оптимальный план - План 3, потому что он использует левостороннее дерево и все выборки и проекции сделаны максимально рано.

### 3.5 Выполнение команды EXPLAIN ANALYSE

```
Hash Join (cost=491.19..5047.40 rows=8676 width=8) (actual time=1.650..10.511 rows=6789 loops=1)
  Hash Cond: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")
    -> Bitmap Heap Scan on "Н_ВЕДОМОСТИ" (cost=490.14..4881.51 rows=26029 width=8) (actual
time=1.484..5.907 rows=26125 loops=1)
      Recheck Cond: ("ИД" > 1250981)
      Heap Blocks: exact=642
      -> Bitmap Index Scan on "ВЕД_ПК" (cost=0.00..483.64 rows=26029 width=0) (actual
time=1.376..1.376 rows=26125 loops=1)
        Index Cond: ("ИД" > 1250981)
    -> Hash (cost=1.04..1.04 rows=1 width=4) (actual time=0.065..0.066 rows=2 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
      -> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=4) (actual
time=0.053..0.056 rows=2 loops=1)
        Filter: (("НАИМЕНОВАНИЕ")::text > 'Ведомость'::text)
        Rows Removed by Filter: 1
Planning Time: 1.005 ms
Execution Time: 10.903 ms
```

## 4 Вывод

При выполнении данной лабораторной работы я узнал понятие индексов в базах данных и как их использовать. Научился составлять планы и выбирать наиболее выгодный. Узнал про виды оптимизации соединения таблиц