

# УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники  
Направление подготовки 09.03.04 Программная инженерия  
Дисциплина «Информационные системы и базы данных»

## **Лабораторная работа №4** *Вариант 313089*

Студент  
*Нуруллаев Д.*  
*P33121*

Преподаватель  
*Шешуков Д.М.*

Санкт-Петербург, 2022 г.

## Описание 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]

## Реализация первого запроса

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

Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,  
Н\_ВЕДОМОСТИ.ДАТА.

Фильтры (AND):

а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ = Ведомость.

б) Н\_ВЕДОМОСТИ.ИД > 1490007.

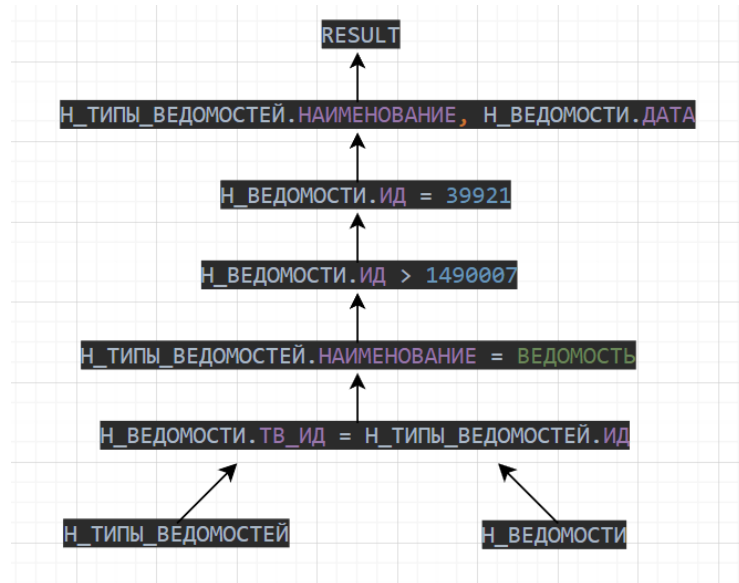
с) Н\_ВЕДОМОСТИ.ИД = 39921.

Вид соединения: LEFT JOIN.

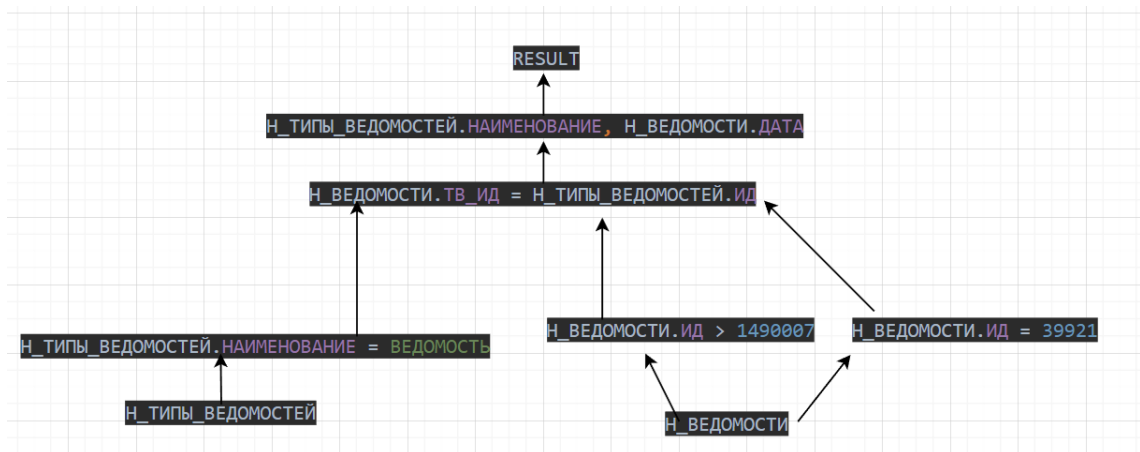
```
SELECT "Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ", "Н_ВЕДОМОСТИ"."ДАТА"  
FROM "Н_ТИПЫ_ВЕДОМОСТЕЙ"  
LEFT OUTER JOIN "Н_ВЕДОМОСТИ" ON "Н_ВЕДОМОСТИ"."ТВ_ИД" =  
"Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД"  
WHERE ("Н_ТИПЫ_ВЕДОМОСТЕЙ"."НАИМЕНОВАНИЕ" = 'ВЕДОМОСТЬ') AND  
("Н_ВЕДОМОСТИ"."ИД" > 1490007) AND ("Н_ВЕДОМОСТИ"."ИД" = 39921))
```

### Планы выполнения:

1)



2)



Во втором плане происходит объединение только нужной выборки, а не всех сущностей. Размер промежуточных данных меньше, значит этот план является оптимальным.

### Индексы:

```
CREATE INDEX ON "H_ТИПЫ_ВЕДОМОСТЕЙ" USING HASH ("ИД");  
CREATE INDEX ON "H_ТИПЫ_ВЕДОМОСТЕЙ" USING BTREE ("НАИМЕНОВАНИЕ");  
CREATE INDEX ON "H_ВЕДОМОСТИ" USING BTREE ("ИД");
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

### Explain Analyse:

Nested Loop (cost=0.42..8.30 rows=1 width=426) (actual time=0.024..0.026 rows=0 loops=1)

Join Filter: ("Н\_ТИПЫ\_ВЕДОМОСТЕЙ"."ИД" = "Н\_ВЕДОМОСТИ"."ТВ\_ИД")

-> Seq Scan on "Н\_ТИПЫ\_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual time=0.024..0.025 rows=0 loops=1)

Filter: (("НАИМЕНОВАНИЕ")::text = 'ВЕДОМОСТЬ'::text)

Rows Removed by Filter: 3

-> Index Scan using "ВЕД\_ПК" on "Н\_ВЕДОМОСТИ" (cost=0.42..7.25 rows=1 width=12) (never executed)

Index Cond: (("ИД" > 1490007) AND ("ИД" = 39921))

Planning Time: 0.234 ms

Execution Time: 0.040 ms

## Реализация второго запроса

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.ЧЛВК\_ИД, Н\_СЕССИЯ.ДАТА.

Фильтры (AND):

а) Н\_ЛЮДИ.ФАМИЛИЯ > Афанасьев.

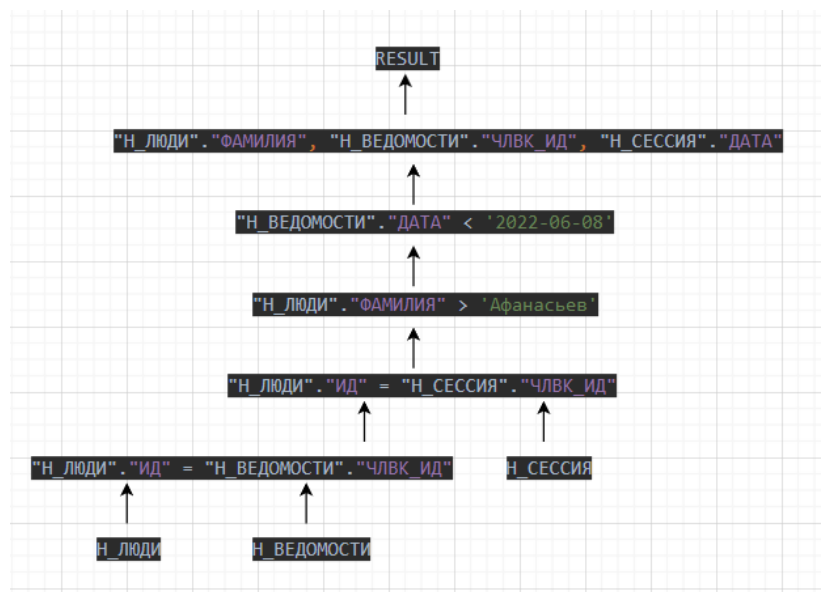
б) Н\_ВЕДОМОСТИ.ДАТА < 2022-06-08.

Вид соединения: LEFT JOIN.

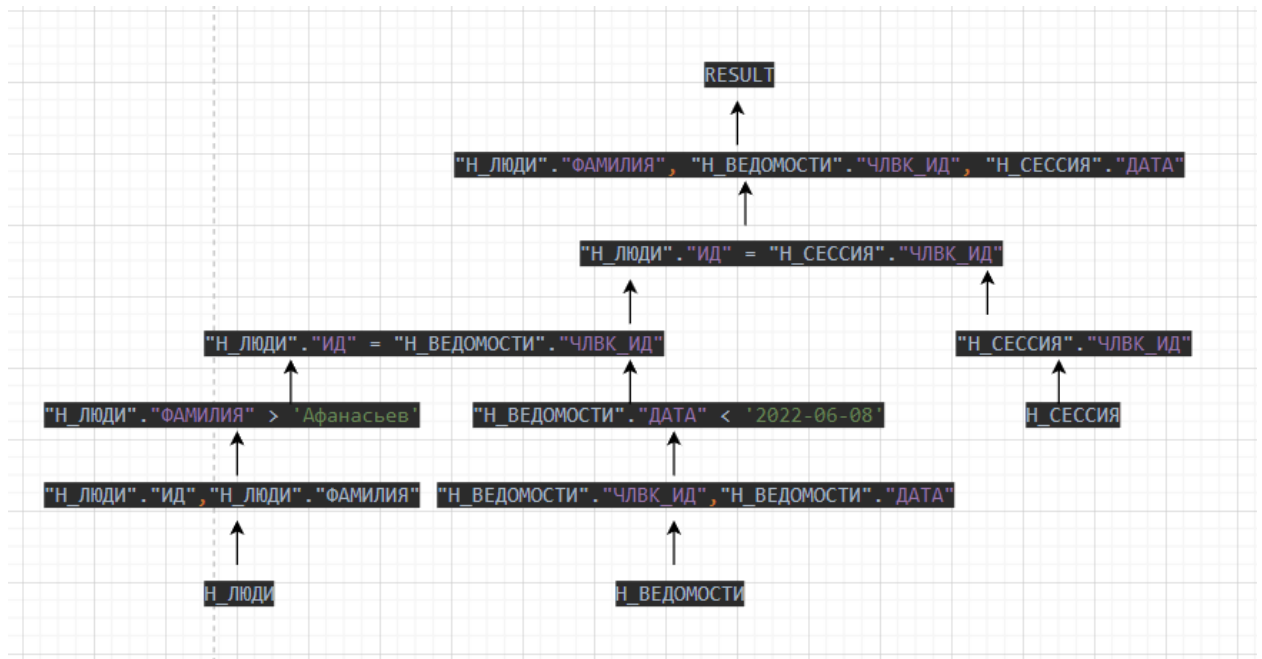
```
SELECT "Н_люди"."ФАМИЛИЯ", "Н_ВЕДОМОСТИ"."ЧЛВК_ИД", "Н_СЕССИЯ"."ДАТА"
FROM "Н_люди"
  LEFT OUTER JOIN "Н_ВЕДОМОСТИ" ON "Н_люди"."ИД" = "Н_ВЕДОМОСТИ"."ЧЛВК_ИД"
  LEFT OUTER JOIN "Н_СЕССИЯ" ON "Н_люди"."ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД"
WHERE (("Н_люди"."ФАМИЛИЯ" > 'Афанасьев') AND ("Н_ВЕДОМОСТИ"."ДАТА" < '2022-06-08'))
```

### Планы выполнения:

1)



2)



Второй план является оптимальным. Из-за того, что выборка происходит на более ранних этапах, идет соединение только нужных атрибутов, и размер промежуточных данных меньше.

*Индексы:*

```
CREATE INDEX ON "Н_ЛЮДИ" USING HASH("ИД");  
CREATE INDEX ON "Н_ЛЮДИ" USING BTREE("ФАМИЛИЯ");  
CREATE INDEX ON "Н_ВЕДОМОСТИ" USING BTREE("ДАТА");
```

Выборка происходит с использованием операторов сравнения, поэтому оптимально использование BTREE. Соединение таблиц с использованием ИД – оптимально использование HASH.

### *Explain Analyse:*

Nested Loop (cost=0.60..4588.02 rows=134487 width=28) (actual time=0.381..6.348 rows=6964 loops=1)  
Join Filter: ("Н\_ЛЮДИ"."ИД" = "Н\_ВЕДОМОСТИ"."ЧЛВК\_ИД")  
-> Nested Loop (cost=0.29..279.41 rows=3095 width=32) (actual time=0.018..2.730 rows=3096 loops=1)  
-> Seq Scan on "Н\_СЕССИЯ" (cost=0.00..108.52 rows=3752 width=12) (actual time=0.005..0.408 rows=3752 loops=1)  
-> Memoize (cost=0.29..0.44 rows=1 width=20) (actual time=0.000..0.000 rows=1 loops=3752)  
Cache Key: "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
Cache Mode: logical  
Hits: 3571 Misses: 181 Evictions: 0 Overflows: 0 Memory Usage: 22kB  
-> Index Scan using "ЧЛВК\_ПК" on "Н\_ЛЮДИ" (cost=0.28..0.43 rows=1 width=20) (actual time=0.003..0.003 rows=1 loops=181)  
Index Cond: ("ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД")  
Filter: (("ФАМИЛИЯ")::text > 'Афанасьев'::text)  
Rows Removed by Filter: 0  
-> Memoize (cost=0.30..6.16 rows=68 width=4) (actual time=0.000..0.001 rows=2 loops=3096)  
Cache Key: "Н\_СЕССИЯ"."ЧЛВК\_ИД"  
Cache Mode: logical  
Hits: 2922 Misses: 174 Evictions: 0 Overflows: 0 Memory Usage: 51kB  
-> Index Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..6.15 rows=68 width=4) (actual time=0.001..0.006 rows=6 loops=174)  
Index Cond: ("ЧЛВК\_ИД" = "Н\_СЕССИЯ"."ЧЛВК\_ИД")  
Filter: ("ДАТА" < '2022-06-08 00:00:00'::timestamp without time zone)  
Planning Time: 0.542 ms  
Execution Time: 6.702 ms

## Выводы

При выполнении лабораторной работы я познакомился с использованием индексов для ускорения обработки запросов в SQL. Изучил, как поэтапно происходит выполнение запроса, какие планы могут использоваться и как индексы влияют на процесс.