

Санкт-Петербургский национальный исследовательский  
университет информационных технологий, механики и оптики.

Кафедра вычислительной техники

Базы данных

**Лабораторная работа №4**

Вариант: **1294**

Выполнил:

Студент группы Р3210

Глушков Дмитрий Сергеевич

Санкт-Петербург  
2018 г.

## Задание

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:  
Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ,  
Н\_ВЕДОМОСТИ.ИД.  
Фильтры (AND):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Экзаменационный лист.  
б) Н\_ВЕДОМОСТИ.ИД = 1250981.  
Вид соединения: INNER JOIN.
2. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.  
Вывести атрибуты: Н\_ЛЮДИ.ОТЧЕСТВО, Н\_ВЕДОМОСТИ.ИД,  
Н\_СЕССИЯ.ЧЛВК\_ИД.  
Фильтры (AND):  
а) Н\_ЛЮДИ.ОТЧЕСТВО = Александрович.  
б) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 105590.  
с) Н\_СЕССИЯ.ДАТА = 2002-01-04.  
Вид соединения: RIGHT JOIN.

### 1. Запрос:

```
SELECT Н_ВЕДОМОСТИ.ИД, Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ
FROM Н_ВЕДОМОСТИ
INNER JOIN Н_ТИПЫ_ВЕДОМОСТЕЙ
ON (Н_ВЕДОМОСТИ.ТВ_ИД = Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД)
WHERE
    Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < 'Экзаменационный лист'
AND
```

```
Н_ВЕДОМОСТИ.ИД = 1250981;
```

### Результат запроса:

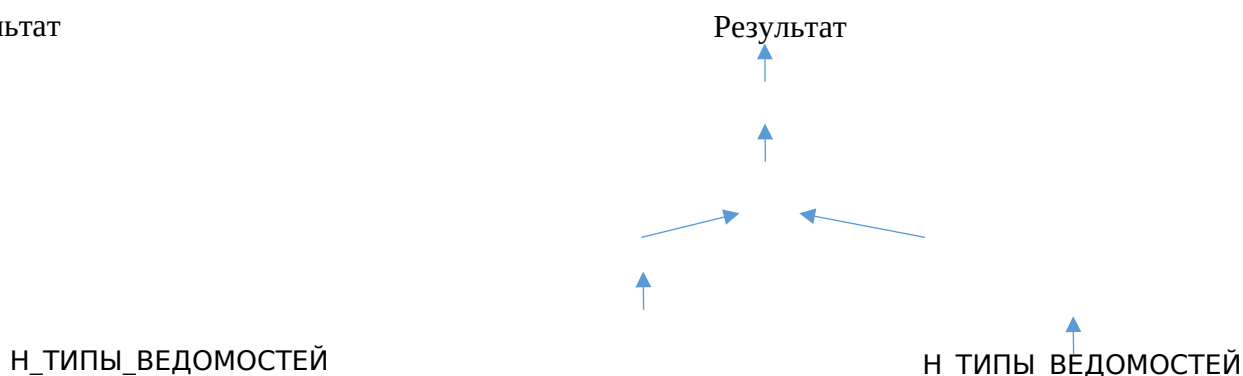
ИД	НАИМЕНОВАНИЕ
1250981	Ведомость

(1 row)

### Планы выполнения запроса:

Из представленных выше планов выполнения запроса наиболее оптимальным является правый план. Это обусловлено тем, что на этапе формирования соединения исходные таблицы уже прошли выборку, а значит это соединение будет содержать меньше строк, и, как следствие, формироваться быстрее.

Результат



### Создание индексов:

```
CREATE INDEX "ВЕДОМОСТИ_ИД_ИНД" ON "Н_ВЕДОМОСТИ" USING HASH("ИД");  
CREATE INDEX "ТИПЫ_ВЕДОМОСТЕЙ_ИД_ИНД" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" USING HASH("ИД");  
CREATE INDEX "ТИПЫ_ВЕДОМОСТЕЙ_НАИМЕНОВАНИЕ_ИНД" ON "Н_ТИПЫ_ВЕДОМОСТЕЙ" USING  
BTREE("НАИМЕНОВАНИЕ");
```

Использование данного набора индексов значительно ускорит время выполнения запроса, так как в этих запросах происходит выборка с использованием операторов сравнения и соединения.

### Вывод команды EXPLAIN ANALYZE:

```
QUERY PLAN  
Nested Loop (cost=0.42..9.49 rows=1 width=422) (actual time=0.025..0.029 rows=1 loops=1)  
  Join Filter: ("Н_ВЕДОМОСТИ"."ТВ_ИД" = "Н_ТИПЫ_ВЕДОМОСТЕЙ"."ИД")  
  Rows Removed by Join Filter: 1  
  -> Index Scan using "ВЕД_ПК" on "Н_ВЕДОМОСТИ" (cost=0.42..8.44 rows=1 width=8)  
      (actual time=0.012..0.013 rows=1 loops=1)
```

```

Index Cond: ("ИД" = 1250981)
-> Seq Scan on "Н_ТИПЫ_ВЕДОМОСТЕЙ" (cost=0.00..1.04 rows=1 width=422) (actual
time=0.010..0.013 rows=2 loops=1)
  Filter: (("НАИМЕНОВАНИЕ")::text < 'Экзаменационный лист'::text)
  Rows Removed by Filter: 1
Planning time: 0.121 ms
Execution time: 0.050 ms
              (10          ows)

```

## 2. Запрос:

```

SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ВЕДОМОСТИ.ИД, Н_СЕССИЯ.ЧЛВК_ИД
FROM Н_ЛЮДИ
RIGHT JOIN Н_ВЕДОМОСТИ ON (Н_ЛЮДИ.ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД)
RIGHT JOIN Н_СЕССИЯ ON (Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД)
WHERE
  Н_ЛЮДИ.ОТЧЕСТВО = 'Александрович'
  AND Н_ВЕДОМОСТИ.ЧЛВК_ИД > 105590
  AND DATE (Н_СЕССИЯ.ДАТА) = '2002-01-04';

```

### Результат запроса:

```

ОТЧЕСТВО | ИД | ЧЛВК_ИД
-----+-----+-----
(0 rows)

```

### Планы выполнения запроса:

Из представленных выше планов выполнения запроса наиболее оптимальным является правый план. Это обусловлено тем, что на этапе формирования соединения исходные таблицы уже прошли выборку, а также была создана промежуточная проекция, а значит это соединение будет содержать меньше строк и столбцов, и, как следствие, формироваться быстрее.

Результат

Результат

### Создание индексов:

```
CREATE INDEX "ВЕДОМОСТИ_ИД_ИНД" ON "Н_ВЕДОМОСТИ" USING HASH("ИД");
CREATE INDEX "ВЕДОМОСТИ_ЧЛВК_ИД_ИНД" ON "Н_ВЕДОМОСТИ" USING BTREE("ЧЛВК_ИД");
CREATE INDEX "ЛЮДИ_ИД_ИНД" ON "Н_ЛЮДИ" USING HASH("ИД");
CREATE INDEX "ЛЮДИ_ОТЧЕСТВО_ИНД" ON "Н_ЛЮДИ" USING HASH("ОТЧЕСТВО");
CREATE INDEX "СЕССИЯ_ДАТА_ИНД" ON "Н_СЕССИЯ" USING HASH("ДАТА");
CREATE INDEX "СЕССИЯ_ЧЛВК_ИД_ИНД" ON "Н_СЕССИЯ" USING HASH("ЧЛВК_ИД");
```

Использование данного набора индексов значительно ускорит время выполнения запроса, так как в этих запросах происходит выборка с использованием операторов сравнения и соединения.

### Вывод команды EXPLAIN ANALYZE:

```
QUERY PLAN
Nested Loop (cost=0.70..315.34 rows=70 width=28) (actual time=0.989..0.989 rows=0
loops=1)
-> Nested Loop (cost=0.28..253.22 rows=2 width=28) (actual time=0.983..0.983 rows=0
loops=1)
-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..127.28 rows=19 width=4) (actual
time=0.131..0.956 rows=7 loops=1)
Filter: (date("ДАТА") = '2002-01-04'::date)
Rows Removed by Filter: 3745
-> Index Scan using "ЧЛВК_ПК" on "Н_ЛЮДИ" (cost=0.28..6.62 rows=1 width=24)
(actual time=0.003..0.003 rows=0 loops=7)
Index Cond: ("ИД" = "Н_СЕССИЯ"."ЧЛВК_ИД")
Filter: (("ОТЧЕСТВО")::text = 'Александрович'::text)
Rows Removed by Filter: 1
-> Index Scan using "ВЕД_ЧЛВК_FK_IFK" on "Н_ВЕДОМОСТИ" (cost=0.42..30.38 rows=68
width=8) (never executed)
Index Cond: (("ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД") AND ("ЧЛВК_ИД" > 105590))
Planning time: 0.536 ms
Execution time: 1.079 ms
(13 rows)
```

### Вывод

В результате выполнения лабораторной работы были изучены и применены на практике методы выполнения SQL запросов, построение планов выполнения запросов, оптимизация процесса выполнения через введение индексов.