

**Университет ИТМО**

**Факультет ПИиКТ**

Лабораторная работа №4 по дисциплине  
“Информационные системы и базы данных”

(3ий курс бакалавриата ФПИиКТ)

**Студент:**

Иванов Евгений Дмитриевич

Группа Р33111

**Преподаватель:**

Харитоновна Анастасия Евгеньевна

## Вариант: 1542

### 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.

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.ИД.  
Фильтры (AND):  
а) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = 3.  
б) Н\_ВЕДОМОСТИ.ИД < 1250981.  
с) Н\_ВЕДОМОСТИ.ИД = 39921.  
Вид соединения: INNER 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.ИД.  
Фильтры: (AND)  
а) Н\_ЛЮДИ.ИМЯ = Роман.  
б) Н\_ОБУЧЕНИЯ.НЗК = 933232.  
Вид соединения: RIGHT JOIN.

## Запросы:

```
select ТВ."ИД", НВ."ИД" from "Н_ТИПЫ_ВЕДОМОСТЕЙ" ТВ
INNER JOIN "Н_ВЕДОМОСТИ" НВ on ТВ."ИД" = НВ."ТВ_ИД"
WHERE ТВ."ИД" = 3 AND НВ."ИД" < 1250981 AND НВ."ИД" =39921;
```

```
select ЛЮДИ."ИМЯ", ОБ."ЧЛВК_ИД", УЧ."ИД" from "Н_ЛЮДИ"
ЛЮДИ
RIGHT JOIN "Н_ОБУЧЕНИЯ" ОБ on ЛЮДИ."ИД" = ОБ."ЧЛВК_ИД"
RIGHT JOIN "Н_УЧЕНИКИ" УЧ on УЧ."ЧЛВК_ИД" = ЛЮДИ."ИД"
WHERE "ИМЯ" = 'Роман' AND ОБ."НЗК" = '933232';
```

## Explain для запросов:

```
Nested Loop (cost=0.42..9.49 rows=1 width=8) (actual
time=0.068..0.069 rows=0 loops=1)
" -> Seq Scan on ""Н_ТИПЫ_ВЕДОМОСТЕЙ"" ""ТВ""
(cost=0.00..1.04 rows=1 width=4) (actual time=0.020..0.021 rows=1
loops=1)"
"      Filter: (""ИД"" = 3)"
      Rows Removed by Filter: 2
" -> Index Scan using ""ВЕД_ПК"" on ""Н_ВЕДОМОСТИ"" ""НВ""
(cost=0.42..8.44 rows=1 width=8) (actual time=0.045..0.045 rows=0
loops=1)"
"      Index Cond: ((""ИД"" < 1250981) AND (""ИД"" = 39921))"
"      Filter: (""ТВ_ИД"" = 3)"
      Rows Removed by Filter: 1
Planning Time: 0.823 ms
Execution Time: 0.143 ms
```

```
Nested Loop (cost=0.57..144.40 rows=5 width=21) (actual
time=0.748..0.749 rows=0 loops=1)
-> Nested Loop (cost=0.28..128.25 rows=1 width=21) (actual
time=0.748..0.749 rows=0 loops=1)
"      -> Seq Scan on ""Н_ОБУЧЕНИЯ"" ""ОБ""
(cost=0.00..119.76 rows=1 width=4) (actual time=0.025..0.725 rows=1
loops=1)"
"      Filter: ((""НЗК"")::text = '933232'::text)"
      Rows Removed by Filter: 5020
```

```

"      -> Index Scan using ""ЧЛВК_РК"" on ""Н_ЛЮДИ"" ""ЛЮДИ""
(cost=0.28..8.30 rows=1 width=17) (actual time=0.020..0.020 rows=0
loops=1)"
"      Index Cond: (""ИД"" = ""ОБ"". ""ЧЛВК_ИД"")"
"      Filter: ((""ИМЯ"")::text = 'Роман'::text)"
"      Rows Removed by Filter: 1
"      -> Index Scan using ""УЧЕН_ОБУЧ_ФК_I"" on ""Н_УЧЕНИКИ""
""УЧ"" (cost=0.29..16.11 rows=5 width=8) (never executed)"
"      Index Cond: (""ЧЛВК_ИД"" = ""ЛЮДИ"". ""ИД"")"
Planning Time: 1.763 ms
Execution Time: 0.823 ms

```

## Индексы:

### Запрос 1:

- 1) Для таблицы Н\_ВЕДОМОСТИ
  - создать индекс типа B-tree для поля “ИД”, так как оно используется для выборки.
- 2) Для таблицы Н\_ТИПЫ\_ВЕДОМОСТЕЙ
  - создание индексов для данной таблицы бесполезно, так как она хранит всего несколько значений(3) и по сути является ENUMом, ну а переборка 3 значений спокойно обходится без индексов.

### Запрос 2:

- 1) Для таблицы Н\_ЛЮДИ
  - создать индекс типа B-tree для поля “ИМЯ”, так как оно используется для выборки.
  - создать индекс типа B-tree для поля “ИД”, так как оно используется для соединения таблиц.
- 2) Для таблицы Н\_ОБУЧЕНИЯ
  - создать индекс типа B-tree для поля “ЧЛВК\_ИД”, так как оно используется для соединения таблиц.
- 3) Для таблицы Н\_УЧЕНИКИ
  - создать индекс типа B-tree для поля “ЧЛВК\_ИД”, так как оно используется для соединения таблиц.

## Планы запросов

1:

a)

```
Н_ТИПЫ_ВЕДОМОСТЕЙ + filter(ид = 3)
Н_ВЕДОМОСТИ + filter(ид = 39921)
Н_ТИПЫ_ВЕДОМОСТЕЙ_R JOIN Н_ВЕДОМОСТИ_R ON
“ИД” = “ТВ_ИД”
JOIN_R + filter(ид(НВ) < 1250981)
RETURN RESULT
```

б)

```
Н_ТИПЫ_ВЕДОМОСТЕЙ + filter(ид = 3)
Н_ВЕДОМОСТИ + filter(ид = 39921)
Н_ВЕДОМОСТИ_R + filter(ид < 1250981)
Н_ТИПЫ_ВЕДОМОСТЕЙ_R JOIN Н_ВЕДОМОСТИ_R ON
“ИД” = “ТВ_ИД”
RETURN RESULT
```

в)

```
Н_ТИПЫ_ВЕДОМОСТЕЙ_R JOIN Н_ВЕДОМОСТИ_R
ON “ИД” = “ТВ_ИД”
JOIN_R + filter(ид(ВЕД) = 39921)
JOIN_R + filter(ид(ТВ) = 3)
JOIN_R + filter(ид(ВЕД) < 1250981)
RETURN RESULT
```

Вывод:

Я бы выбрал вариант Б, так как мы начинаем накладывать фильтры на обе таблицы, после чего их соединяем без получения лишних данных в результате. Если бы мы использовали индексацию, то фильтры проводились бы через данную таблицу, после чего она бы соединялась с необходимыми данными.

2:

а)

```
Н_ЛЮДИ + filter(ИМЯ = 'Роман')
Н_ОБУЧЕНИЯ + filter(НЗК = '933232')
Н_ЛЮДИ JOIN Н_ОБУЧЕНИЯ
ON "ИД" = "ЧЛВК_ИД"
JOIN_R JOIN Н_УЧЕНИКИ
ON "ИД" = "ЧЛВК_ИД"
RETURN RESULT
```

б)

```
Н_ЛЮДИ + filter(ИМЯ = 'Роман')
Н_ОБУЧЕНИЯ + filter(НЗК = '933232')
Н_ЛЮДИ JOIN Н_ОБУЧЕНИЯ
ON "ИД" = "ЧЛВК_ИД"
JOIN_R JOIN Н_УЧЕНИКИ
ON "ИД" = "ЧЛВК_ИД"
RETURN RESULT
```

Вывод:

Я считаю что вариант А лучше, так как в нём мы сначала применяем фильтрацию, а потом соединяем таблицы и таким образом избавляемся от получения лишних данных.

При индексации происходили бы сортировки и соединения без обращения к каким-либо столбцам, после чего данные для вывода добавлялись по внутренним ссылкам и происходил return.

**Выводы:** в ходе лабораторной работы я изучил индексы баз данных, познакомился с планами выполнения запросов.

