МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

«Базы данных»

Вариант № 4582

***Выполнил:***Студент группы P3116  
 Григорьев Даниил Александрович

***Преподаватель:***Пименов Данила Дмитриевич

# 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).

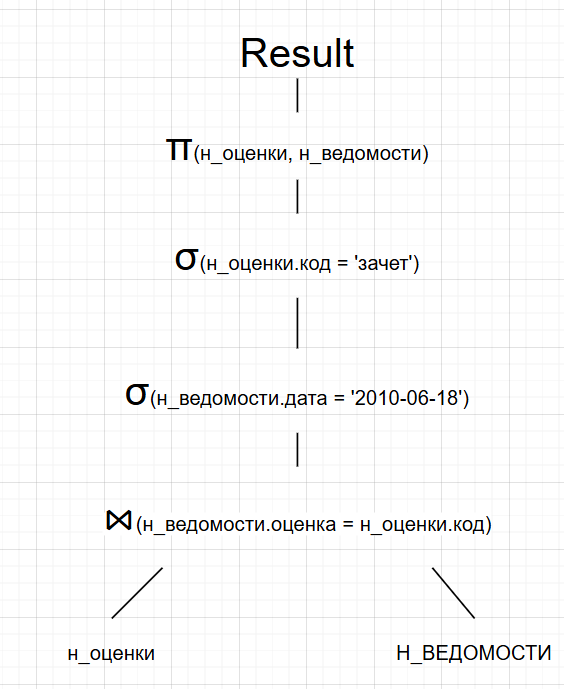
1. Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:  
   Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
   Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД.  
   Фильтры (AND):  
   a) Н\_ОЦЕНКИ.КОД = зачет.  
   b) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.  
   Вид соединения: 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.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  
   Фильтры: (AND)  
   a) Н\_ЛЮДИ.ОТЧЕСТВО = Сергеевич.  
   b) Н\_ОБУЧЕНИЯ.НЗК = 933232.  
   c) Н\_УЧЕНИКИ.НАЧАЛО > 1996-09-01.  
   Вид соединения: INNER JOIN.

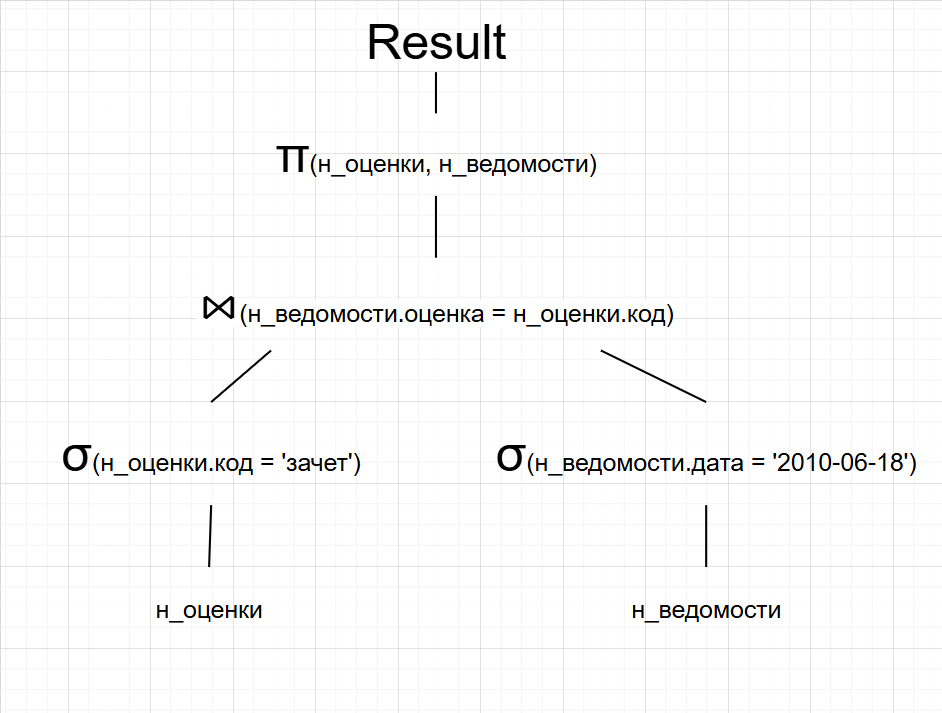
**Решение**

Запрос №1

Таблицы: Н\_ОЦЕНКИ, Н\_ВЕДОМОСТИ.  
Вывести атрибуты: Н\_ОЦЕНКИ.КОД, Н\_ВЕДОМОСТИ.ИД.  
Фильтры (AND):  
a) Н\_ОЦЕНКИ.КОД = зачет.  
b) Н\_ВЕДОМОСТИ.ДАТА = 2010-06-18.  
Вид соединения: INNER JOIN.

select "Н\_ОЦЕНКИ".КОД, "Н\_ВЕДОМОСТИ".ИД  
from "Н\_ОЦЕНКИ"  
join "Н\_ВЕДОМОСТИ" on "Н\_ВЕДОМОСТИ"."ОЦЕНКА" = "Н\_ОЦЕНКИ"."КОД"  
where "Н\_ОЦЕНКИ"."КОД" = 'зачет' and "Н\_ВЕДОМОСТИ"."ДАТА" = '2010-06-18'



Оптимальным является план номер 2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

**Индексы:**

-- Индекс на КОД в Н\_ОЦЕНКИ  
CREATE INDEX "ИНДЕКС\_ОЦЕНКИ\_КОД"  
ON "Н\_ОЦЕНКИ"  
USING hash("КОД");  
  
-- Индекс на ДАТА в Н\_ВЕДОМОСТИ  
CREATE INDEX "ИНДЕКС\_ВЕДОМОСТИ\_ДАТА"  
ON "Н\_ВЕДОМОСТИ"  
USING btree("ДАТА");  
  
-- Индекс на ОЦЕНКА в Н\_ВЕДОМОСТИ (для соединения)  
CREATE INDEX "ИНДЕКС\_ВЕДОМОСТИ\_ОЦЕНКА"  
ON "Н\_ВЕДОМОСТИ"  
USING hash("ОЦЕНКА");

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходят выборки и соединения с использованием операторов сравнения.

* Для поля "КОД" используется **hash**, так как фильтр равенства = 'зачет'.
* Для поля "ДАТА" используется **btree**, так как возможны операции сравнения, но здесь фильтр = '2010-06-18' тоже работает эффективно.
* Для поля "ОЦЕНКА" также используется **hash**, так как соединение выполняется по условию равенства.

При добавлении индексов план выполнения изменится, так как произойдёт индексный поиск вместо последовательного сканирования, и **Hash Join** станет быстрее благодаря подготовленным хеш-таблицам.

Explain analyze:

Nested Loop (cost=0.29..218.45 rows=34 width=38) (actual time=0.044..0.091 rows=12 loops=1)

" -> Seq Scan on ""Н\_ОЦЕНКИ"" (cost=0.00..1.11 rows=1 width=34) (actual time=0.015..0.016 rows=1 loops=1)"

" Filter: ((""КОД"")::text = 'зачет'::text)"

Rows Removed by Filter: 8

" -> Index Scan using ""ВЕД\_ДАТА\_I"" on ""Н\_ВЕДОМОСТИ"" (cost=0.29..217.00 rows=34 width=10) (actual time=0.026..0.070 rows=12 loops=1)"

" Index Cond: (""ДАТА"" = '2010-06-18 00:00:00'::timestamp without time zone)"

" Filter: ((""ОЦЕНКА"")::text = 'зачет'::text)"

Rows Removed by Filter: 129

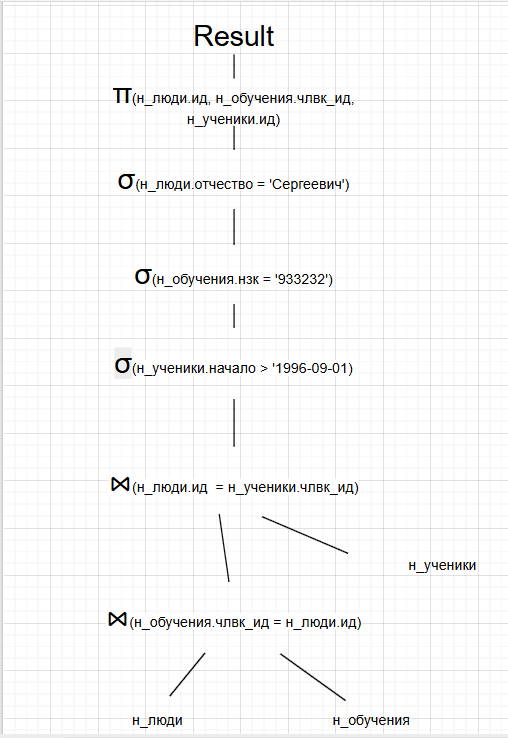
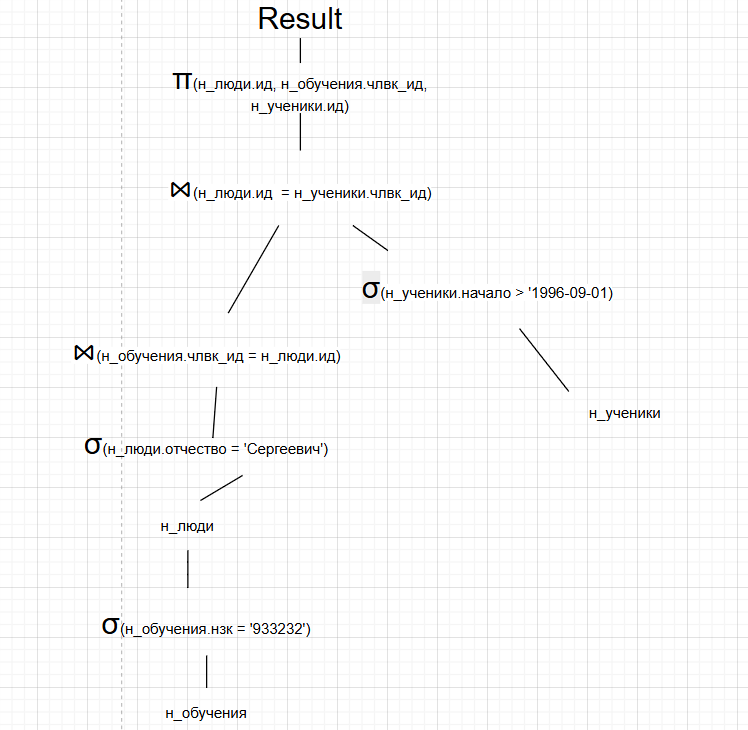
Planning Time: 0.163 ms

Execution Time: 0.117 ms

Запрос №2

Таблицы: Н\_ЛЮДИ, Н\_ОБУЧЕНИЯ, Н\_УЧЕНИКИ.  
Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД.  
Фильтры: (AND)  
a) Н\_ЛЮДИ.ОТЧЕСТВО = Сергеевич.  
b) Н\_ОБУЧЕНИЯ.НЗК = 933232.  
c) Н\_УЧЕНИКИ.НАЧАЛО > 1996-09-01.  
Вид соединения: INNER JOIN.

SELECT Н\_ЛЮДИ.ИД, Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД, Н\_УЧЕНИКИ.ИД  
FROM Н\_ЛЮДИ  
JOIN Н\_ОБУЧЕНИЯ ON Н\_ЛЮДИ.ИД = Н\_ОБУЧЕНИЯ.ЧЛВК\_ИД  
JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД  
WHERE Н\_ЛЮДИ.ОТЧЕСТВО = 'Сергеевич'  
 AND Н\_ОБУЧЕНИЯ.НЗК = '933232'  
 AND Н\_УЧЕНИКИ.НАЧАЛО > '1996-09-01';



Оптимальным является план номер 2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

Индекс:

-- Индекс на поле ОТЧЕСТВО в таблице Н\_ЛЮДИ  
CREATE INDEX "ИНДЕКС\_ЛЮДИ\_ОТЧЕСТВО"  
ON "Н\_ЛЮДИ" USING hash("ОТЧЕСТВО");  
  
-- Индекс на поле НЗК в таблице Н\_ОБУЧЕНИЯ  
CREATE INDEX "ИНДЕКС\_ОБУЧЕНИЯ\_НЗК"  
ON "Н\_ОБУЧЕНИЯ" USING hash("НЗК");  
  
-- Индекс на поле НАЧАЛО в таблице Н\_УЧЕНИКИ  
CREATE INDEX "ИНДЕКС\_УЧЕНИКИ\_НАЧАЛО"  
ON "Н\_УЧЕНИКИ" USING btree("НАЧАЛО");  
  
-- Индексы на поля соединения (необязательно, но ускоряют JOIN)  
CREATE INDEX "ИНДЕКС\_ОБУЧЕНИЯ\_ЧЛВК\_ИД"  
ON "Н\_ОБУЧЕНИЯ" USING hash("ЧЛВК\_ИД");  
  
CREATE INDEX "ИНДЕКС\_УЧЕНИКИ\_ЧЛВК\_ИД"  
ON "Н\_УЧЕНИКИ" USING hash("ЧЛВК\_ИД");

Добавление этих индексов должно ускорить выполнение запроса, так как по перечисленным полям (ОТЧЕСТВО, НЗК, НАЧАЛО, ЧЛВК\_ИД) происходит выборка с использованием операторов сравнения.

* Для поля **ОТЧЕСТВО** в таблице Н\_ЛЮДИ используется фильтр равенства (=), поэтому эффективнее использовать **hash-индекс**.
* Для поля **НЗК** в таблице Н\_ОБУЧЕНИЯ также используется равенство (=), поэтому применяется **hash-индекс**.
* Для поля **НАЧАЛО** в таблице Н\_УЧЕНИКИ используется сравнение (>), поэтому здесь целесообразно использовать **btree-индекс**.
* Для полей соединений (**ЧЛВК\_ИД**) создание **hash-индексов** ускорит выполнение соединений между таблицами.

При добавлении этих индексов планы выполнения запросов изменятся:

* **Index Scan** заменит **Seq Scan**, что ускорит выборку.
* Соединение между таблицами будет происходить быстрее благодаря наличию индексов на ключевых полях.
* **Nested Loop Join** или **Hash Join** станет быстрее, так как PostgreSQL сможет эффективно использовать созданные индексы для поиска совпадений.

Explain analyze:

Nested Loop (cost=0.57..134.72 rows=4 width=12) (actual time=0.049..0.564 rows=2 loops=1)

-> Nested Loop (cost=0.28..128.11 rows=1 width=8) (actual time=0.022..0.520 rows=1 loops=1)

" -> Seq Scan on ""Н\_ОБУЧЕНИЯ"" (cost=0.00..119.76 rows=1 width=4) (actual time=0.012..0.509 rows=1 loops=1)"

" Filter: ((""НЗК"")::text = '933232'::text)"

Rows Removed by Filter: 5020

" -> Index Scan using ""ЧЛВК\_PK"" on ""Н\_ЛЮДИ"" (cost=0.28..8.30 rows=1 width=4) (actual time=0.007..0.007 rows=1 loops=1)"

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

" Filter: ((""ОТЧЕСТВО"")::text = 'Сергеевич'::text)"

" -> Index Scan using ""УЧЕН\_ОБУЧ\_FK\_I"" on ""Н\_УЧЕНИКИ"" (cost=0.29..6.56 rows=5 width=8) (actual time=0.026..0.042 rows=2 loops=1)"

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

" Filter: (""НАЧАЛО"" > '1996-09-01 00:00:00'::timestamp without time zone)"

Rows Removed by Filter: 1

Planning Time: 1.369 ms

Execution Time: 0.602 ms