

Национальный исследовательский университет ИТМО

Факультет Программной Инженерии и Компьютерной техники

Лабораторная работа №4 Вариант №3155

Выполнил:

Валиев Руслан Новруз оглы

Группа Р3107

Преподаватель:

Вербовой Александр Александрович

Санкт-Петербург
2025

Оглавление

Задание:	3
Запрос №1:	4
Индексы:	4
Результат EXPLAIN ANALYSE:	5
Запрос №2:	6
Индексы:	6
Результат EXPLAIN ANALYSE:	7
Вывод:	8

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?

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

Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н_ЛЮДИ.ФАМИЛИЯ > Афанасьев.

б) Н_СЕССИЯ.УЧГОД > 2008/2009.

с) Н_СЕССИЯ.УЧГОД = 2003/2004.

Вид соединения: LEFT JOIN.

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

Таблицы: Н_ЛЮДИ, Н_ОБУЧЕНИЯ, Н_УЧЕНИКИ.

Вывести атрибуты: Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.

Фильтры: (AND)

а) Н_ЛЮДИ.ИД > 100865.

б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163484.

Вид соединения: RIGHT JOIN.

Запрос №1:

Таблицы: Н_ЛЮДИ, Н_СЕССИЯ.

Вывести атрибуты: Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.УЧГОД.

Фильтры (AND):

а) Н_ЛЮДИ.ФАМИЛИЯ > Афанасьев.

б) Н_СЕССИЯ.УЧГОД > 2008/2009.

с) Н_СЕССИЯ.УЧГОД = 2003/2004.

Вид соединения: LEFT JOIN.

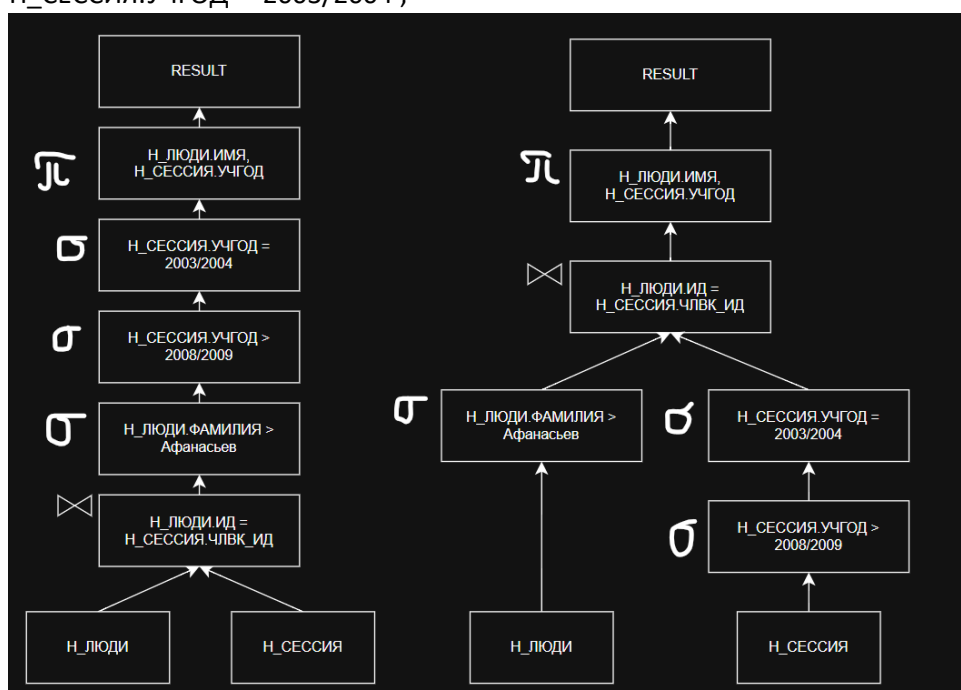
SELECT Н_ЛЮДИ.ИМЯ, Н_СЕССИЯ.УЧГОД

FROM Н_ЛЮДИ

LEFT JOIN Н_СЕССИЯ ON Н_ЛЮДИ.ИД = Н_СЕССИЯ.ЧЛВК_ИД

WHERE Н_ЛЮДИ.ФАМИЛИЯ > 'Афанасьев' AND Н_СЕССИЯ.УЧГОД > '2008/2009' OR

Н_СЕССИЯ.УЧГОД = '2003/2004';



Оптимальным является план №2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

Индексы:

```
CREATE INDEX "ИНДЕКС_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING hash("ИД");  
CREATE INDEX "ИНДЕКС_СЕССИЯ_ЧЛВК_ИД" ON "Н_СЕССИЯ" USING hash("ЧЛВК_ИД");  
CREATE INDEX "ИНДЕКС_СЕССИЯ_УЧГОД" ON "Н_СЕССИЯ" USING btree("УЧГОД");  
CREATE INDEX "ИНДЕКС_ЛЮДИ_ФАМИЛИЯ" ON "Н_ЛЮДИ" USING btree("ФАМИЛИЯ");
```

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения. Так же быстрее будет происходить соединение таблиц. В последних двух случаях

используются операторы сравнения „>“ и „<“, так что эффективнее использовать btree. В первых двух случаях используется прямое сравнение, так что эффективнее использовать хэш-индекс.

Результат EXPLAIN ANALYSE:

Hash Join (cost=215.16..346.01 rows=1135 width=23) (actual time=1.859..4.739 rows=1145 loops=1)

Hash Cond: ("Н_СЕССИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")

Join Filter: (((("Н_ЛЮДИ"."ФАМИЛИЯ")::text > 'Афанасьев'::text) AND ("Н_СЕССИЯ"."УЧГОД")::text > '2008/2009'::text)) OR (("Н_СЕССИЯ"."УЧГОД")::text = '2003/2004'::text))

Rows Removed by Join Filter: 41

-> Seq Scan on "Н_СЕССИЯ" (cost=0.00..127.28 rows=1361 width=14) (actual time=0.014..1.693 rows=1456 loops=1)

Filter: (((("УЧГОД")::text > '2008/2009'::text) OR (("УЧГОД")::text = '2003/2004'::text))

Rows Removed by Filter: 2296

-> Hash (cost=151.18..151.18 rows=5118 width=33) (actual time=1.822..1.823 rows=5118 loops=1)

Buckets: 8192 Batches: 1 Memory Usage: 400kB

-> Seq Scan on "Н_ЛЮДИ" (cost=0.00..151.18 rows=5118 width=33) (actual time=0.005..0.938 rows=5118 loops=1)

Planning Time: 0.541 ms

Execution Time: 4.830 ms

Запрос №2:

Таблицы: Н_ЛЮДИ, Н_ОБУЧЕНИЯ, Н_УЧЕНИКИ.

Вывести атрибуты: Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД.

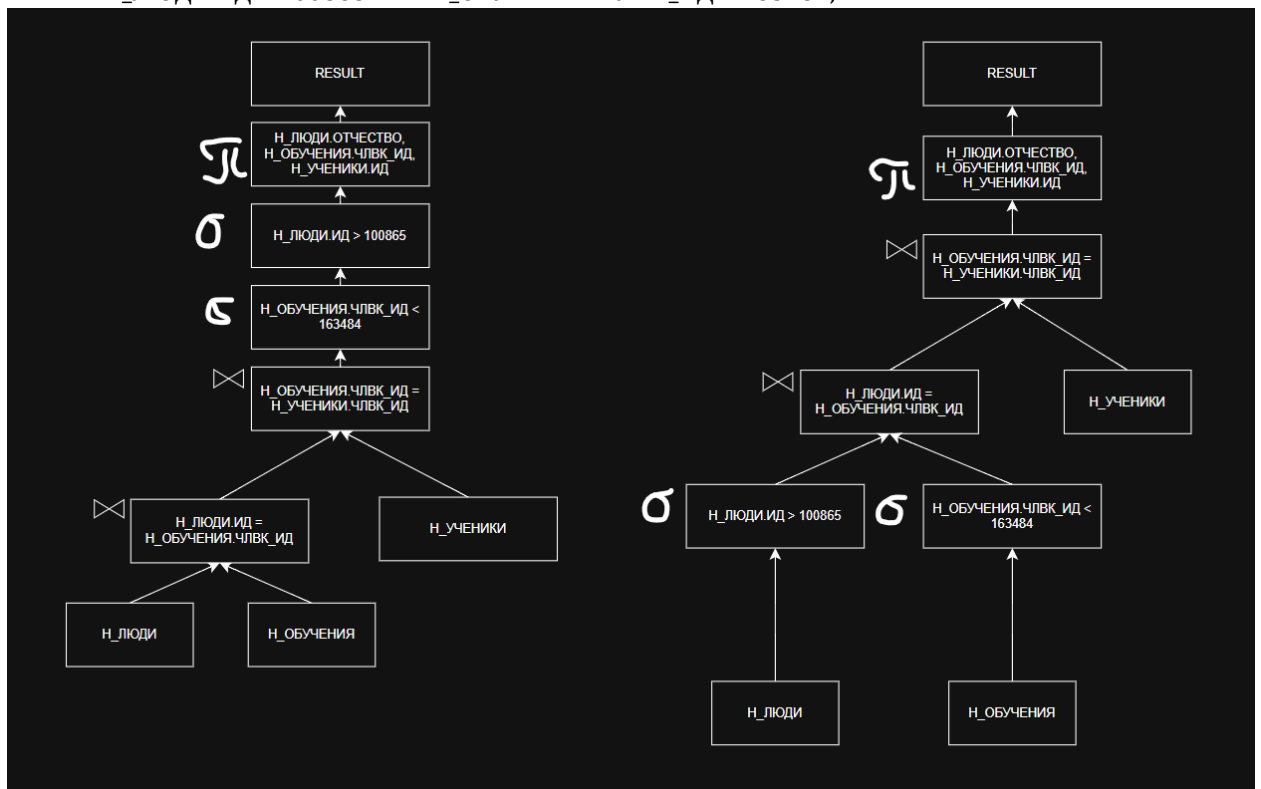
Фильтры: (AND)

а) Н_ЛЮДИ.ИД > 100865.

б) Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163484.

Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ОТЧЕСТВО, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.ИД
FROM Н_ЛЮДИ
RIGHT JOIN Н_ОБУЧЕНИЯ
ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
RIGHT JOIN Н_УЧЕНИКИ
ON Н_ОБУЧЕНИЯ.ЧЛВК_ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
WHERE Н_ЛЮДИ.ИД > 100865 AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163484;
```



Оптимальным является план №2, так как он производит объединение таблиц по ранее выбранным атрибутам, а не по таблицам целиком.

Индексы:

```
CREATE INDEX "ИНДЕКС_ЛЮДИ_ИД" ON "Н_ЛЮДИ" USING btree ("ИД");
CREATE INDEX "ИНДЕКС_ОБУЧЕНИЯ_ЧЛВК_ИД" ON "Н_ОБУЧЕНИЯ" USING btree ("ЧЛВК_ИД");
CREATE INDEX "ИНДЕКС_УЧЕНИКИ_ЧЛВК_ИД" ON "Н_УЧЕНИКИ" USING hash ("ЧЛВК_ИД");
```

Добавление этих индексов должно ускорить выполнение запросов, так как по перечисленным полям происходит выборка с использованием оператора сравнения. Так же быстрее будет происходить соединение таблиц.

Результат EXPLAIN ANALYSE:

Hash Join (cost=422.99..1546.97 rows=22746 width=28) (actual time=5.582..17.696 rows=23635 loops=1)

Hash Cond: ("Н_УЧЕНИКИ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")

-> Seq Scan on "Н_УЧЕНИКИ" (cost=0.00..809.11 rows=23311 width=8) (actual time=0.007..2.449 rows=23311 loops=1)

-> Hash (cost=360.56..360.56 rows=4994 width=28) (actual time=5.548..5.552 rows=5020 loops=1)

Buckets: 8192 Batches: 1 Memory Usage: 371kB

-> Hash Join (cost=227.61..360.56 rows=4994 width=28) (actual time=1.869..4.343 rows=5020 loops=1)

Hash Cond: ("Н_ОБУЧЕНИЯ"."ЧЛВК_ИД" = "Н_ЛЮДИ"."ИД")

-> Seq Scan on "Н_ОБУЧЕНИЯ" (cost=0.00..119.76 rows=5020 width=4) (actual time=0.007..0.882 rows=5020 loops=1)

Filter: ("ЧЛВК_ИД" < 163484)

Rows Removed by Filter: 1

-> Hash (cost=163.97..163.97 rows=5091 width=24) (actual time=1.846..1.848 rows=5077 loops=1)

Buckets: 8192 Batches: 1 Memory Usage: 354kB

-> Seq Scan on "Н_ЛЮДИ" (cost=0.00..163.97 rows=5091 width=24) (actual time=0.004..1.021 rows=5077 loops=1)

Filter: ("ИД" > 100865)

Rows Removed by Filter: 41

Planning Time: 0.604 ms

Execution Time: 18.844 ms

Вывод: во время выполнения данной лабораторной работы я научился оптимизировать запросы, составлять наиболее выгодный план выполнения запросов, используя для этого подходящие виды индексов.