Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3

Индексы

тема

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | А.К. Погребников |
|  | подпись, дата | инициалы, фамилия |
| Студент КИ18-16б 031831229 |  | В.А. Прекель |
| номер группы, зачетной книжки | подпись, дата | инициалы, фамилия |

Красноярск 2020

# 1 Цель работы

Добавить индексы в таблицы.

# 2 Общая постановка задачи

Добавить индексы в таблицы.

# 3 Исходный код

Листинг 1 – Тестируемый запрос

SELECT "SupportQuestions"."SupportTicketId",  
 true "IsQuestion",  
 "SupportQuestions"."SendTimestamp",  
 "SupportQuestions"."ReadTimestamp",  
 C."FirstName",  
 C."LastName",  
 "SupportQuestions"."Text"  
FROM public."SupportQuestions"  
 JOIN "SupportTickets" ST on ST."SupportTicketId" = "SupportQuestions"."SupportTicketId"  
 JOIN "Customers" C on C."CustomerId" = ST."CustomerId"  
WHERE C."FirstName" = 'Юлия'  
UNION  
SELECT "SupportAnswers"."SupportTicketId",  
 false,  
 "SupportAnswers"."SendTimestamp",  
 NULL,  
 SO."FirstName",  
 SO."LastName",  
 "SupportAnswers"."Text"  
FROM public."SupportAnswers"  
 JOIN "SupportOperators" SO on SO."SupportOperatorId" = "SupportAnswers"."SupportOperatorId"  
WHERE SO."FirstName" = 'Юлия'  
ORDER BY "SupportTicketId", "SendTimestamp";

Листинг 2 – Добавление и удаление индексов

CREATE INDEX ON public."SupportOperators" ("FirstName");  
CREATE INDEX ON public."Customers" ("FirstName");  
  
DROP INDEX "SupportOperators\_FirstName\_idx";  
DROP INDEX "Customers\_FirstName\_idx";

Листинг 3 – Explain Analyze без индексов

Sort (cost=25667.47..25678.24 rows=4309 width=329) (actual time=79.551..79.737 rows=3999 loops=1)  
" Sort Key: ""SupportAnswers"".""SupportTicketId"", ""SupportAnswers"".""SendTimestamp"""  
 Sort Method: quicksort Memory: 679kB  
 -> HashAggregate (cost=25364.26..25407.35 rows=4309 width=329) (actual time=77.289..78.174 rows=3999 loops=1)  
" Group Key: ""SupportAnswers"".""SupportTicketId"", (false), ""SupportAnswers"".""SendTimestamp"", (NULL::timestamp with time zone), so.""FirstName"", so.""LastName"", ""SupportAnswers"".""Text"""  
 -> Gather (cost=1000.84..25288.86 rows=4309 width=329) (actual time=4.281..99.984 rows=3999 loops=1)  
 Workers Planned: 2  
 Workers Launched: 2  
 -> Parallel Append (cost=0.84..23857.96 rows=3787 width=329) (actual time=4.851..48.265 rows=1333 loops=3)  
 -> Parallel Hash Join (cost=2899.38..5921.10 rows=1052 width=81) (actual time=4.708..12.714 rows=565 loops=3)  
" Hash Cond: (""SupportAnswers"".""SupportOperatorId"" = so.""SupportOperatorId"")"  
" -> Parallel Seq Scan on ""SupportAnswers"" (cost=0.00..2766.86 rows=97086 width=48) (actual time=0.005..3.908 rows=55015 loops=3)"  
 -> Parallel Hash (cost=2889.18..2889.18 rows=816 width=32) (actual time=4.590..4.590 rows=448 loops=3)  
 Buckets: 2048 Batches: 1 Memory Usage: 112kB  
" -> Parallel Seq Scan on ""SupportOperators"" so (cost=0.00..2889.18 rows=816 width=32) (actual time=0.015..13.489 rows=1343 loops=1)"  
" Filter: ((""FirstName"")::text = 'Юлия'::text)"  
 Rows Removed by Filter: 126657  
 -> Nested Loop (cost=0.84..17874.83 rows=1050 width=88) (actual time=0.336..53.207 rows=1153 loops=2)  
 -> Nested Loop (cost=0.42..17477.59 rows=700 width=32) (actual time=0.320..50.250 rows=772 loops=2)  
" -> Parallel Seq Scan on ""Customers"" c (cost=0.00..13746.46 rows=2721 width=32) (actual time=0.159..44.458 rows=2853 loops=2)"  
" Filter: ((""FirstName"")::text = 'Юлия'::text)"  
 Rows Removed by Filter: 288658  
" -> Index Scan using ""IX\_SupportTickets\_CustomerId"" on ""SupportTickets"" st (cost=0.42..1.36 rows=1 width=8) (actual time=0.002..0.002 rows=0 loops=5706)"  
" Index Cond: (""CustomerId"" = c.""CustomerId"")"  
" -> Index Scan using ""IX\_SupportQuestions\_SupportTicketId"" on ""SupportQuestions"" (cost=0.42..0.55 rows=2 width=59) (actual time=0.003..0.003 rows=1 loops=1543)"  
" Index Cond: (""SupportTicketId"" = st.""SupportTicketId"")"  
Planning Time: 0.468 ms  
Execution Time: 105.198 ms

Листинг 4 – Explain Analyze с индексами

Sort (cost=20141.81..20152.58 rows=4309 width=329) (actual time=51.517..51.627 rows=3999 loops=1)  
" Sort Key: ""SupportQuestions"".""SupportTicketId"", ""SupportQuestions"".""SendTimestamp"""  
 Sort Method: quicksort Memory: 679kB  
 -> HashAggregate (cost=19838.61..19881.70 rows=4309 width=329) (actual time=49.422..50.231 rows=3999 loops=1)  
" Group Key: ""SupportQuestions"".""SupportTicketId"", (true), ""SupportQuestions"".""SendTimestamp"", ""SupportQuestions"".""ReadTimestamp"", c.""FirstName"", c.""LastName"", ""SupportQuestions"".""Text"""  
 -> Append (cost=10078.32..19763.20 rows=4309 width=329) (actual time=5.273..47.594 rows=3999 loops=1)  
 -> Nested Loop (cost=10078.32..13881.02 rows=2521 width=88) (actual time=5.272..27.154 rows=2305 loops=1)  
 -> Hash Join (cost=10077.90..12927.65 rows=1680 width=32) (actual time=5.258..23.491 rows=1543 loops=1)  
" Hash Cond: (st.""CustomerId"" = c.""CustomerId"")"  
" -> Seq Scan on ""SupportTickets"" st (cost=0.00..2456.00 rows=150000 width=8) (actual time=0.008..7.454 rows=149999 loops=1)"  
 -> Hash (cost=9996.28..9996.28 rows=6530 width=32) (actual time=5.218..5.218 rows=5706 loops=1)  
 Buckets: 8192 Batches: 1 Memory Usage: 388kB  
" -> Bitmap Heap Scan on ""Customers"" c (cost=159.03..9996.28 rows=6530 width=32) (actual time=1.300..4.421 rows=5706 loops=1)"  
" Recheck Cond: ((""FirstName"")::text = 'Юлия'::text)"  
 Heap Blocks: exact=4227  
" -> Bitmap Index Scan on ""Customers\_FirstName\_idx"" (cost=0.00..157.40 rows=6530 width=0) (actual time=0.937..0.937 rows=5706 loops=1)"  
" Index Cond: ((""FirstName"")::text = 'Юлия'::text)"  
" -> Index Scan using ""IX\_SupportQuestions\_SupportTicketId"" on ""SupportQuestions"" (cost=0.42..0.55 rows=2 width=59) (actual time=0.002..0.002 rows=1 loops=1543)"  
" Index Cond: (""SupportTicketId"" = st.""SupportTicketId"")"  
 -> Hash Join (cost=1937.81..5817.54 rows=1788 width=81) (actual time=1.118..20.232 rows=1694 loops=1)  
" Hash Cond: (""SupportAnswers"".""SupportOperatorId"" = so.""SupportOperatorId"")"  
" -> Seq Scan on ""SupportAnswers"" (cost=0.00..3446.47 rows=165047 width=48) (actual time=0.021..8.682 rows=165046 loops=1)"  
 -> Hash (cost=1920.48..1920.48 rows=1387 width=32) (actual time=1.082..1.082 rows=1343 loops=1)  
 Buckets: 2048 Batches: 1 Memory Usage: 98kB  
" -> Bitmap Heap Scan on ""SupportOperators"" so (cost=35.17..1920.48 rows=1387 width=32) (actual time=0.364..0.930 rows=1343 loops=1)"  
" Recheck Cond: ((""FirstName"")::text = 'Юлия'::text)"  
 Heap Blocks: exact=955  
" -> Bitmap Index Scan on ""SupportOperators\_FirstName\_idx"" (cost=0.00..34.82 rows=1387 width=0) (actual time=0.289..0.289 rows=1343 loops=1)"  
" Index Cond: ((""FirstName"")::text = 'Юлия'::text)"  
Planning Time: 2.364 ms  
Execution Time: 52.089 ms