

Smetanin Gleb. Week 5.

Exercise 1.

Without indexes (Query 1):

The screenshot shows the PostgreSQL Query Editor interface. The query being executed is: `EXPLAIN ANALYZE SELECT * FROM new_crowd WHERE (length(review) >= 40 AND length(review) <= 150 AND length(review) % 3 = 0)`. The execution plan is displayed under the 'Data Output' tab. The plan shows a 'Parallel Seq Scan on new_crowd' with a cost of 0.00..4649.65 and an actual time of 0.033..203.605. The filter applied is `((length(review) >= 40) AND (length(review) <= 150) AND ((length(review) % 3) = 0))`, which removed 41566 rows. The planning time was 1.305 ms and the execution time was 318.163 ms. A green status bar at the bottom indicates 'Successfully run. Total query'.

Step	Operation	Cost	Actual Time	Rows	Width	Loops
1	Gather	1000.00..5649.85	0.483..315.921	16869	211	1
2	Workers Planned:			1		
3	Workers Launched:			1		
4	Parallel Seq Scan on new_crowd	0.00..4649.65	0.033..203.605	8435	211	2
5	Filter: ((length(review) >= 40) AND (length(review) <= 150) AND ((length(review) % 3) = 0))					
6	Rows Removed by Filter:			41566		
7	Planning Time:		1.305 ms			
8	Execution Time:		318.163 ms			

BTree (Query 1):

The screenshot shows the PostgreSQL Query Editor interface. The query being executed is: `EXPLAIN ANALYZE SELECT * FROM new_crowd WHERE (length(review) >= 40 AND length(review) <= 150 AND length(review) % 3 = 0)`. The execution plan is displayed under the 'Data Output' tab. The plan shows an 'Index Scan using "BTree" on new_crowd' with a cost of 0.41..12.44 and an actual time of 0.456..43.244. The planning time was 2.813 ms and the execution time was 44.610 ms.

Step	Operation	Cost	Actual Time	Rows	Width	Loops
1	Index Scan using "BTree" on new_crowd	0.41..12.44	0.456..43.244	16869	211	1
2	Planning Time:		2.813 ms			
3	Execution Time:		44.610 ms			

Without indexes (Query 2 and payments table):

The screenshot shows the PostgreSQL Query Editor interface. The query editor contains the following SQL query:

```
1 EXPLAIN ANALYZE SELECT * FROM payment WHERE amount = 7.99
```

The query is executed, and the results are displayed in the 'Data Output' tab. The results show the query plan for the query:

QUERY PLAN
1 Seq Scan on payment (cost=0.00..290.45 rows=621 width=26) (actual time=0.036..6.824 rows=621 loops=1)
2 Filter: (amount = 7.99)
3 Rows Removed by Filter: 13975
4 Planning Time: 0.209 ms
5 Execution Time: 6.896 ms

Hash (Query 2 and payments table):

The screenshot shows the PostgreSQL Query Editor interface. The query editor contains the following SQL query:

```
1 EXPLAIN ANALYZE SELECT * FROM payment WHERE amount = 7.99
```

The query is executed, and the results are displayed in the 'Data Output' tab. The results show the query plan for the query:

QUERY PLAN
1 Bitmap Heap Scan on payment (cost=24.81..140.58 rows=621 width=26) (actual time=0.119..1.234 rows=621 loops=1)
2 Recheck Cond: (amount = 7.99)
3 Heap Blocks: exact=106
4 -> Bitmap Index Scan on "Hash" (cost=0.00..24.66 rows=621 width=0) (actual time=0.086..0.087 rows=621 loops=1)
5 Index Cond: (amount = 7.99)
6 Planning Time: 2.697 ms
7 Execution Time: 1.360 ms

Conclusion: Indexes allow query execute faster but use additional memory.