

NAME -JATAN SAHU
ID-202218061

IT667 - Database Management Systems

Lab Assignment 8 - Indexing

SET SEARCH_PATH TO 'TPCH_DB';

QUERY :1









BEFORE INDEXING :-

SELECT * FROM supplier ;

EXPLAIN ANALYZE

SELECT * FROM SUPPLIER WHERE s_nationkey=7;

Data output Messages Notifications

Data output	
       	
	<div>QUERY PLAN</div> <div>text</div> <div>🔒</div>
1	Seq Scan on supplier (cost=0.00..323.00 rows=10000 width=145) (actual time=0.013..0.717 rows=10000 loops=1)
2	Planning time: 0.050 ms
3	Execution time: 1.484 ms

Total rows: 3 of 3 Query complete 00:00:00.037

AFTER INDEXING :-

EXPLAIN ANALYZE

SELECT * FROM SUPPLIER WHERE s_nationkey=7;

CREATE INDEX IDX_SUPPLIER_NATION_KEY ON SUPPLIER (S_NATIONKEY);

Data output		Messages	Notifications
<div><div><div>≡+</div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>			
	QUERY PLAN		
	text		
1	Bitmap Heap Scan on supplier (cost=11.35..244.94 rows=396 width=145) (actual time=0.064..0.173 rows...		
2	Recheck Cond: (s_nationkey = 7)		
3	Heap Blocks: exact=188		
4	-> Bitmap Index Scan on idx_supplier_nation_key (cost=0.00..11.25 rows=396 width=0) (actual time=0.04...		
5	Index Cond: (s_nationkey = 7)		
6	Planning time: 0.792 ms		
7	Execution time: 0.205 ms		

Total rows: 7 of 7	Query complete 00:00:00.038
--------------------	-----------------------------

2. BEFORE INDEXING ANALYSIS : -

SELECT * FROM CUSTOMER;

EXPLAIN ANALYZE

SELECT * FROM CUSTOMER WHERE c_nationkey=16;

Data output Messages Notifications

QUERY PLAN

text

1

Seq Scan on customer (cost=0.00..5478.00 rows=5745 width=160) (actual time=0.015..11.564 rows=5974 loops=1)

2

Filter: (c_nationkey = 16)

3

Rows Removed by Filter: 144026

4

Planning time: 0.072 ms

5

Execution time: 11.723 ms

Total rows: 5 of 5

Query complete 00:00:00.083

AFTER

```
SELECT * FROM CUSTOMER;
```

```
EXPLAIN ANALYZE
```

```
SELECT * FROM CUSTOMER WHERE c_nationkey=16;
```

```
CREATE INDEX IDX_CUSTOMER_NATIONKEY ON CUSTOMER (C_NATIONKEY);
```

Data output		Messages	Notifications
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	QUERY PLAN text <div>🔒</div>		
1	Bitmap Heap Scan on customer (cost=108.94..3934.28 rows=5745 width=160) (actual time=0.712..2.370 rows=...		
2	Recheck Cond: (c_nationkey = 16)		
3	Heap Blocks: exact=2954		
4	-> Bitmap Index Scan on idx_customer_nationkey (cost=0.00..107.51 rows=5745 width=0) (actual time=0.499..0...		
5	Index Cond: (c_nationkey = 16)		
6	Planning time: 0.714 ms		
7	Execution time: 2.525 ms		
Total rows: 7 of 7		Query complete 00:00:00.056	

3. BEFORE INDEXING ANALYSIS : -

SET SEARCH_PATH TO 'TPCH_DB'

EXPLAIN ANALYZE







```

SELECT PS_PARTKEY, SUM(PS_SUPPLYCOST*PS_AVAILQTY) AS VALUE
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY AND S_NATIONKEY = N_NATIONKEY AND N_NAME
= 'GERMANY'
GROUP BY PS_PARTKEY
HAVING SUM(PS_SUPPLYCOST*PS_AVAILQTY) > (SELECT
SUM(PS_SUPPLYCOST*PS_AVAILQTY) *
0.0001000000
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY AND S_NATIONKEY = N_NATIONKEY AND N_NAME
= 'GERMANY')
ORDER BY VALUE DESC;

```


AFTER INDEXING : -

CREATE INDEX idx_n_nationkey on NATION(N_NATIONKEY)

Data output Messages Notifications		
     		
	QUERY PLAN	
	text	
30	-> Gather (cost=1360.03..26828.69 rows=32000 width=14) (actual time=8.711..568.841 rows=31680 loops=1)	
31	Workers Planned: 2	
32	Workers Launched: 2	
33	-> Hash Join (cost=360.02..22628.69 rows=13333 width=14) (actual time=16.376..360.693 rows=10560 loops=3)	
34	Hash Cond: (partsupp.ps_suppley = supplier.s_suppley)	
35	-> Parallel Seq Scan on partsupp (cost=0.00..20885.33 rows=333333 width=18) (actual time=6.096..239.933 rows=266667 loops=3)	
36	-> Hash (cost=355.02..355.02 rows=400 width=4) (actual time=9.724..9.726 rows=396 loops=3)	
37	Buckets: 1024 Batches: 1 Memory Usage: 22kB	
38	-> Hash Join (cost=1.33..355.02 rows=400 width=4) (actual time=1.438..9.494 rows=396 loops=3)	
39	Hash Cond: (supplier.s_nationkey = nation.n_nationkey)	
40	-> Seq Scan on supplier (cost=0.00..323.00 rows=10000 width=8) (actual time=0.683..4.898 rows=10000 loops=3)	
41	-> Hash (cost=1.31..1.31 rows=1 width=4) (actual time=0.670..0.671 rows=1 loops=3)	
42	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
43	-> Seq Scan on nation (cost=0.00..1.31 rows=1 width=4) (actual time=0.645..0.652 rows=1 loops=3)	
44	Filter: (n_name = 'GERMANY')::bpchar)	
45	Rows Removed by Filter: 24	
46	Planning time: 2.498 ms	
47	Execution time: 1274.185 ms	
Total rows: 47 of 47		Query complete 00:00:01.325 Ln 15, Col 1

4.BEFORE INDEXING : -

```
SELECT C_COUNT, COUNT(*) AS CUSTDIST FROM (SELECT C_CUSTKEY,
COUNT(O_ORDERKEY) FROM CUSTOMER left outer join ORDERS on C_CUSTKEY =
O_CUSTKEY AND O_COMMENT not like '%%special%%requests%%' GROUP BY
C_CUSTKEY) AS C_ORDERS (C_CUSTKEY, C_COUNT) GROUP BY C_COUNT ORDER BY
CUSTDIST DESC, C_COUNT DESC
```

Data output		Messages	Notifications
<div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div>			
	QUERY PLAN		
	text		
8	-> Gather Merge (cost=171965.54..212780.02 rows=300000 width=12) (actual time=6203.484..10580.446 rows=150000 loops=1)		
9	Workers Planned: 2		
10	Workers Launched: 2		
11	-> Partial GroupAggregate (cost=170965.52..177152.55 rows=150000 width=12) (actual time=5857.957..6409.064 rows=50000 loops=3)		
12	Group Key: customer.c_custkey		
13	-> Sort (cost=170965.52..172527.86 rows=624938 width=8) (actual time=5857.941..6075.759 rows=511308 loops=3)		
14	Sort Key: customer.c_custkey		
15	Sort Method: external merge Disk: 10432kB		
16	-> Hash Left Join (cost=69650.13..102257.76 rows=624938 width=8) (actual time=3804.944..5324.154 rows=511308 loops=3)		
17	Hash Cond: (customer.c_custkey = orders.o_custkey)		
18	-> Parallel Seq Scan on customer (cost=0.00..4228.00 rows=62500 width=4) (actual time=1.110..65.916 rows=50000 loops=3)		
19	-> Hash (cost=45043.00..45043.00 rows=1499850 width=8) (actual time=3771.837..3771.838 rows=1483918 loops=3)		
20	Buckets: 131072 Batches: 32 Memory Usage: 2863kB		
21	-> Seq Scan on orders (cost=0.00..45043.00 rows=1499850 width=8) (actual time=3.047..2562.421 rows=1483918 loops=3)		
22	Filter: ((o_comment)::text !~ '%special%requests%':text)		
23	Rows Removed by Filter: 16082		
24	Planning time: 0.925 ms		
25	Execution time: 11029.641 ms		
Total rows: 25 of 25		Query complete 00:00:11.650	
		Ln 2, Col 1	

AFTER INDEXING :-

CREATE INDEX idx_orderkey on ORDERS(O_ORDERKEY)

Data output		Messages	Notifications
<div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div>			
	QUERY PLAN		
	text		
8	-> Gather Merge (cost=171965.54..212780.02 rows=300000 width=12) (actual time=6339.263..6880.059 rows=150000 loops=1)		
9	Workers Planned: 2		
10	Workers Launched: 2		
11	-> Partial GroupAggregate (cost=170965.52..177152.55 rows=150000 width=12) (actual time=5703.986..6003.179 rows=50000 loops=3)		
12	Group Key: customer.c_custkey		
13	-> Sort (cost=170965.52..172527.86 rows=624938 width=8) (actual time=5703.973..5842.995 rows=511308 loops=3)		
14	Sort Key: customer.c_custkey		
15	Sort Method: external merge Disk: 11352kB		
16	-> Hash Left Join (cost=69650.13..102257.76 rows=624938 width=8) (actual time=3867.599..5226.359 rows=511308 loops=3)		
17	Hash Cond: (customer.c_custkey = orders.o_custkey)		
18	-> Parallel Seq Scan on customer (cost=0.00..4228.00 rows=62500 width=4) (actual time=0.023..27.929 rows=50000 loops=3)		
19	-> Hash (cost=45043.00..45043.00 rows=1499850 width=8) (actual time=3834.565..3834.566 rows=1483918 loops=3)		
20	Buckets: 131072 Batches: 32 Memory Usage: 2863kB		
21	-> Seq Scan on orders (cost=0.00..45043.00 rows=1499850 width=8) (actual time=0.949..2549.827 rows=1483918 loops=3)		
22	Filter: ((o_comment)::text !~ '%special%requests%':text)		
23	Rows Removed by Filter: 16082		
24	Planning time: 2.207 ms		
25	Execution time: 7069.564 ms		
Total rows: 25 of 25		Query complete 00:00:07.097	Ln 2, Col 1

5. BEFORE INDEXING :-

EXPLAIN ANALYZE

SELECT SUM(L_EXTENDEDPRICE*L_DISCOUNT) AS REVENUE
FROM LINEITEM

WHERE L_SHIPDATE >= '1994-01-01' AND L_SHIPDATE <DATE '1994-01-01' + INTERVAL '1
YEAR'
AND L_DISCOUNT BETWEEN .06 - 0.01 AND .06 + 0.01 AND L_QUANTITY < 24;

Data output		Messages	Notifications
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div></div> </div>			
	QUERY PLAN		
	text		
1	Finalize Aggregate (cost=170926.17..170926.18 rows=1 width=32) (actual time=5283.999..5320.298 rows=1 loops=1)		
2	-> Gather (cost=170925.95..170926.16 rows=2 width=32) (actual time=5272.280..5320.274 rows=3 loops=1)		
3	Workers Planned: 2		
4	Workers Launched: 2		
5	-> Partial Aggregate (cost=169925.95..169925.96 rows=1 width=32) (actual time=5087.741..5087.742 rows=1 loops=3)		
6	-> Parallel Seq Scan on lineitem (cost=0.00..169686.57 rows=47876 width=12) (actual time=7.950..5024.783 rows=38053 loops=3)		
7	Filter: ((l_shipdate >= '1994-01-01'::date) AND (l_shipdate < '1995-01-01 00:00:00'::timestamp without time zone) AND (l_discount >= 0.05) AND (l_discount <= 0.07) AND ...)		
8	Rows Removed by Filter: 1962352		
9	Planning time: 0.411 ms		
10	Execution time: 5320.407 ms		

Total rows: 10 of 10

Query complete 00:00:05.380

Ln 7, Col 6

AFTER INDEXING

CREATE INDEX idx_l_shipdate on LINEITEM(L_SHIPDATE)



QUERY PLAN	
	text
1	Finalize Aggregate (cost=142831.21..142831.22 rows=1 width=32) (actual time=2485.095..2551.365 rows=1 loops=1)
2	-> Gather (cost=142830.99..142831.20 rows=2 width=32) (actual time=2481.783..2551.338 rows=3 loops=1)
3	Workers Planned: 2
4	Workers Launched: 2
5	-> Partial Aggregate (cost=141830.99..141831.00 rows=1 width=32) (actual time=2115.714..2115.715 rows=1 loops=3)
6	-> Parallel Bitmap Heap Scan on lineitem (cost=19473.65..141591.61 rows=47876 width=12) (actual time=190.533..2062.581 rows=38053 loops=3)
7	Recheck Cond: ((l_shipdate >= '1994-01-01'::date) AND (l_shipdate < '1995-01-01 00:00:00'::timestamp without time zone))
8	Rows Removed by Index Recheck: 961142
9	Filter: ((l_discount >= 0.05) AND (l_discount <= 0.07) AND (l_quantity < '24'::numeric))
10	Rows Removed by Filter: 265098
11	Heap Blocks: exact=14016 lossy=23579
12	-> Bitmap Index Scan on idx_l_shipdate (cost=0.00..19444.92 rows=927249 width=0) (actual time=533.051..533.052 rows=909455 loops=1)
13	Index Cond: ((l_shipdate >= '1994-01-01'::date) AND (l_shipdate < '1995-01-01 00:00:00'::timestamp without time zone))
14	Planning time: 2.074 ms
15	Execution time: 2551.484 ms

Total rows: 15 of 15 Query complete 00:00:02.587

Ln 2, Col 1