Matriculation Number: A0149874L

Fabian Pascal - Question 3b (Plans)

[1a]
Before:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Join (cost=189.11..480.70 rows=9 width=26) (actual time=1.061..2.702 rows=7 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.008..1.091 rows=10000 loops=1)
4	Filter: (empid IS NOT NULL)
5	-> Hash (cost=189.00..189.00 rows=9 width=10) (actual time=0.744..0.745 rows=7 loops=1)
6	Buckets: 1024 Batches: 1 Memory Usage: 9kB
7	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9 width=10) (actual time=0.067..0.741 rows=7 loops=1)
8	Filter: (salary = 189170)
9	Rows Removed by Filter: 9993
10	Planning Time: 0.072 ms
11	Execution Time: 2.722 ms

After:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Nested Loop (cost=4.64..105.09 rows=9 width=26) (actual time=0.018..0.045 rows=9 loops=1)
2	-> Bitmap Heap Scan on payroll pay (cost=4.35..30.34 rows=9 width=10) (actual time=0.010..0.015 rows=9 loops=1)
3	Recheck Cond: (salary = 189170)
4	Heap Blocks: exact=6
5	-> Bitmap Index Scan on payroll_salary_index (cost=0.00..4.35 rows=9 width=0) (actual time=0.006..0.006 rows=9 loops=1)
6	Index Cond: (salary = 189170)
7	-> Index Scan using employee_pkey on employee per (cost=0.29..8.30 rows=1 width=26) (actual time=0.003..0.003 rows=1 loops=9)
8	Index Cond: ((empid = pay.empid) AND (empid IS NOT NULL))
9	Planning Time: 0.194 ms
10	Execution Time: 0.072 ms

[1b]
Before:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Semi Join (cost=189.11..469.46 rows=9 width=26) (actual time=1.021..3.231 rows=7 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.019..1.029 rows=10000 loops=1)
4	-> Hash (cost=189.00..189.00 rows=9 width=10) (actual time=0.811..0.812 rows=7 loops=1)
5	Buckets: 1024 Batches: 1 Memory Usage: 9kB
6	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9 width=10) (actual time=0.121..0.806 rows=7 loops=1)
7	Filter: (salary = 189170)
8	Rows Removed by Filter: 9993
9	Planning Time: 0.113 ms
10	Execution Time: 3.257 ms

After:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Nested Loop (cost=4.64..105.07 rows=9 width=26) (actual time=0.017..0.045 rows=9 loops=1)
2	-> Bitmap Heap Scan on payroll pay (cost=4.35..30.34 rows=9 width=10) (actual time=0.009..0.015 rows=9 loops=1)
3	Recheck Cond: (salary = 189170)
4	Heap Blocks: exact=6
5	-> Bitmap Index Scan on payroll_salary_index (cost=0.00..4.35 rows=9 width=0) (actual time=0.005..0.005 rows=9 loops=1)
6	Index Cond: (salary = 189170)
7	-> Index Scan using employee_pkey on employee per (cost=0.29..8.30 rows=1 width=26) (actual time=0.003..0.003 rows=1 loops=9)
8	Index Cond: (empid = pay.empid)
9	Planning Time: 0.155 ms
10	Execution Time: 0.066 ms

[1c]
Before:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Semi Join (cost=189.11..469.46 rows=9 width=26) (actual time=1.245..3.799 rows=7 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.007..1.188 rows=10000 loops=1)
4	-> Hash (cost=189.00..189.00 rows=9 width=10) (actual time=1.036..1.037 rows=7 loops=1)
5	Buckets: 1024 Batches: 1 Memory Usage: 9kB
6	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9 width=10) (actual time=0.072..1.027 rows=7 loops=1)
7	Filter: (salary = 189170)
8	Rows Removed by Filter: 9993
9	Planning Time: 0.094 ms
10	Execution Time: 3.824 ms

After:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Nested Loop (cost=4.64..105.07 rows=9 width=26) (actual time=0.043..0.177 rows=9 loops=1)
2	-> Bitmap Heap Scan on payroll pay (cost=4.35..30.34 rows=9 width=10) (actual time=0.035..0.110 rows=9 loops=1)
3	Recheck Cond: (salary = 189170)
4	Heap Blocks: exact=6
5	-> Bitmap Index Scan on payroll_salary_index (cost=0.00..4.35 rows=9 width=0) (actual time=0.031..0.031 rows=9 loops=1)
6	Index Cond: (salary = 189170)
7	-> Index Scan using employee_pkey on employee per (cost=0.29..8.30 rows=1 width=26) (actual time=0.006..0.006 rows=1 loops=9)
8	Index Cond: (empid = pay.empid)
9	Planning Time: 0.182 ms
10	Execution Time: 0.232 ms

[1d]
Before:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Join (cost=189.11..480.70 rows=9 width=26) (actual time=0.844..2.349 rows=7 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.007..0.707 rows=10000 loops=1)
4	-> Hash (cost=189.00..189.00 rows=9 width=10) (actual time=0.693..0.694 rows=7 loops=1)
5	Buckets: 1024 Batches: 1 Memory Usage: 9kB
6	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9 width=10) (actual time=0.064..0.689 rows=7 loops=1)
7	Filter: (salary = 189170)
8	Rows Removed by Filter: 9993
9	Planning Time: 0.082 ms
10	Execution Time: 2.368 ms

After:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Nested Loop (cost=4.64..105.07 rows=9 width=26) (actual time=0.015..0.039 rows=9 loops=1)
2	-> Bitmap Heap Scan on payroll pay (cost=4.35..30.34 rows=9 width=10) (actual time=0.008..0.013 rows=9 loops=1)
3	Recheck Cond: (salary = 189170)
4	Heap Blocks: exact=6
5	-> Bitmap Index Scan on payroll_salary_index (cost=0.00..4.35 rows=9 width=0) (actual time=0.005..0.005 rows=9 loops=1)
6	Index Cond: (salary = 189170)
7	-> Index Scan using employee_pkey on employee per (cost=0.29..8.30 rows=1 width=26) (actual time=0.002..0.002 rows=1 loops=9)
8	Index Cond: (empid = pay.empid)
9	Planning Time: 0.226 ms
10	Execution Time: 0.056 ms

[1e]
Before:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Anti Join (cost=313.89..605.47 rows=9 width=26) (actual time=3.343..6.808 rows=7 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.006..1.057 rows=10000 loops=1)
4	-> Hash (cost=189.00..189.00 rows=9991 width=10) (actual time=3.038..3.039 rows=9993 loops=1)
5	Buckets: 16384 Batches: 1 Memory Usage: 538kB
6	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9991 width=10) (actual time=0.005..1.535 rows=9993 loops=1)
7	Filter: (salary <> 189170)
8	Rows Removed by Filter: 7
9	Planning Time: 0.076 ms
10	Execution Time: 6.844 ms

A0149874L
After:

cs4221-fabian-pascal-redo/postgres@PostgreSQL 13

Query Editor | Query History | Data Output | Notifications | Messages | Explain

	QUERY PLAN text
1	Hash Anti Join (cost=313.89..605.47 rows=9 width=26) (actual time=3.228..5.978 rows=9 loops=1)
2	Hash Cond: (per.empid = pay.empid)
3	-> Seq Scan on employee per (cost=0.00..254.00 rows=10000 width=26) (actual time=0.008..0.895 rows=10000 loops=1)
4	-> Hash (cost=189.00..189.00 rows=9991 width=10) (actual time=2.810..2.811 rows=9991 loops=1)
5	Buckets: 16384 Batches: 1 Memory Usage: 538kB
6	-> Seq Scan on payroll pay (cost=0.00..189.00 rows=9991 width=10) (actual time=0.005..1.451 rows=9991 loops=1)
7	Filter: (salary <> 189170)
8	Rows Removed by Filter: 9
9	Planning Time: 0.264 ms
10	Execution Time: 6.012 ms

References used for Qn 1a - 1e:

- https://www.youtube.com/watch?v=QaRVoXOyob8
- https://www.youtube.com/watch?v=8OCAxk1Rybg
- https://www.tutorialspoint.com/sql/sql-indexes.htm
- https://www.youtube.com/watch?v=fsG1XaZEa78