KOREA UNIVERSITY
DATAX LAB

Chapter 15 – Lab Solution

Query Processing

Exercise 1.a Answer

- SELECT * FROM supplier, nation WHERE s_nationkey=n_nationkey;
- Hash join

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, nation WHERE s_nationkey=n_nationkey;
                                                          QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=442.00..1024.40 rows=27000 width=261) (actual time=1.943..3.473 rows=9580 loops=1)
   Hash Cond: (nation.n_nationkey = supplier.s_nationkey)
   ->  Seq Scan on nation  (cost=0.00..15.40 rows=540 width=122) (actual time=0.008..0.009 rows=26 loops=1)
   ->  Hash  (cost=317.00..317.00 rows=10000 width=139) (actual time=1.895..1.895 rows=10000 loops=1)
         Buckets: 16384  Batches: 1  Memory Usage: 1848kB
         ->  Seq Scan on supplier  (cost=0.00..317.00 rows=10000 width=139) (actual time=0.004..0.549 rows=10000 loops=1)
 Planning Time: 0.077 ms
 Execution Time: 3.867 ms
(8개 행)
```

Exercise 1.b Answer

- SELECT * FROM supplier, nation WHERE s_nationkey=n_nationkey ORDER BY s_nationkey;
- Merge join (sort)

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, nation WHERE s_nationkey=n_nationkey ORDER BY s_nationkey;
                                                              QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------
 Merge Join  (cost=1021.29..1428.99 rows=27000 width=261) (actual time=2.170..4.325 rows=9580 loops=1)
   Merge Cond: (nation.n_nationkey = supplier.s_nationkey)
   ->  Sort  (cost=39.91..41.26 rows=540 width=122) (actual time=0.015..0.016 rows=25 loops=1)
         Sort Key: nation.n_nationkey
         Sort Method: quicksort  Memory: 26kB
         ->  Seq Scan on nation  (cost=0.00..15.40 rows=540 width=122) (actual time=0.008..0.010 rows=26 loops=1)
   ->  Sort  (cost=981.39..1006.39 rows=10000 width=139) (actual time=2.085..2.799 rows=10000 loops=1)
         Sort Key: supplier.s_nationkey
         Sort Method: quicksort  Memory: 2873kB
         ->  Seq Scan on supplier  (cost=0.00..317.00 rows=10000 width=139) (actual time=0.004..0.555 rows=10000 loops=1)
 Planning Time: 0.077 ms
 Execution Time: 4.734 ms
(12개 행)
```

Exercise 2.a Answer

- SELECT * FROM nation as A, nation as B WHERE A.n_nationkey=B.n_nationkey;
- Hash join

```
postgres=# EXPLAIN ANALYZE SELECT * FROM nation as A, nation as B WHERE A.n_nationkey=B.n_nationkey;
                                                      QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=22.15..89.93 rows=1458 width=244) (actual time=0.022..0.028 rows=26 loops=1)
   Hash Cond: (a.n_nationkey = b.n_nationkey)
   ->  Seq Scan on nation a  (cost=0.00..15.40 rows=540 width=122) (actual time=0.009..0.010 rows=26 loops=1)
   ->  Hash  (cost=15.40..15.40 rows=540 width=122) (actual time=0.009..0.009 rows=26 loops=1)
         Buckets: 1024  Batches: 1  Memory Usage: 10kB
         ->  Seq Scan on nation b  (cost=0.00..15.40 rows=540 width=122) (actual time=0.003..0.005 rows=26 loops=1)
 Planning Time: 0.058 ms
 Execution Time: 0.045 ms
(8개 행)
```

Exercise 2.b Answer

- SELECT * FROM nation as A, nation as B WHERE A.n_nationkey>B.n_nationkey;
- Nested loop join

```
postgres=# EXPLAIN ANALYZE SELECT * FROM nation as A, nation as B WHERE A.n_nationkey>B.n_nationkey;
                                                      QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..4406.15 rows=97200 width=244) (actual time=0.020..0.090 rows=325 loops=1)
   Join Filter: (a.n_nationkey > b.n_nationkey)
   Rows Removed by Join Filter: 351
   ->  Seq Scan on nation a  (cost=0.00..15.40 rows=540 width=122) (actual time=0.007..0.009 rows=26 loops=1)
   ->  Materialize  (cost=0.00..18.10 rows=540 width=122) (actual time=0.000..0.001 rows=26 loops=26)
         ->  Seq Scan on nation b  (cost=0.00..15.40 rows=540 width=122) (actual time=0.003..0.005 rows=26 loops=1)
 Planning Time: 0.046 ms
 Execution Time: 0.109 ms
(8개 행)
```

Exercise 3 Answer

- SELECT * FROM supplier, table1 WHERE s_suppkey=sorted;
- SELECT * FROM supplier, table1 WHERE s_suppkey=unsorted;
- Hash join
- Execution time: sorted ≈ unsorted

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_suppkey=sorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=443.00..229786.17 rows=145860 width=192) (actual time=2.393..1191.057 rows=50000 loops=1)
   Hash Cond: (table1.sorted = supplier.s_suppkey)
   ->  Seq Scan on table1  (cost=0.00..203093.00 rows=10000000 width=53) (actual time=0.034..664.120 rows=10000000 loops=1)
   ->  Hash  (cost=318.00..318.00 rows=10000 width=139) (actual time=2.306..2.307 rows=10000 loops=1)
         Buckets: 16384  Batches: 1  Memory Usage: 1848kB
         ->  Seq Scan on supplier  (cost=0.00..318.00 rows=10000 width=139) (actual time=0.006..0.812 rows=10000 loops=1)
 Planning Time: 0.121 ms
 Execution Time: 1192.049 ms
(8개 행)


postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_suppkey=unsorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=443.00..229786.07 rows=56025 width=192) (actual time=2.214..1296.471 rows=50164 loops=1)
   Hash Cond: (table1.unsorted = supplier.s_suppkey)
   ->  Seq Scan on table1  (cost=0.00..203093.00 rows=10000000 width=53) (actual time=0.030..668.649 rows=10000000 loops=1)
   ->  Hash  (cost=318.00..318.00 rows=10000 width=139) (actual time=2.072..2.073 rows=10000 loops=1)
         Buckets: 16384  Batches: 1  Memory Usage: 1848kB
         ->  Seq Scan on supplier  (cost=0.00..318.00 rows=10000 width=139) (actual time=0.006..0.796 rows=10000 loops=1)
 Planning Time: 0.527 ms
 Execution Time: 1298.166 ms
(8개 행)
```

Exercise 4 Answer

- SELECT * FROM supplier, table1 WHERE s_suppkey=sorted;
- SELECT * FROM supplier, table1 WHERE s_suppkey=unsorted;
- Merge join (index)
- Execution time: sorted < unsorted

```
postgres=# CREATE INDEX sorted_idx on table1(sorted);
CREATE INDEX
postgres=# CREATE INDEX unsorted_idx on table1(unsorted);
CREATE INDEX
postgres=# CREATE INDEX suppkey_idx on supplier(s_suppkey);
CREATE INDEX
```

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_suppkey=sorted;
                                                              QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------
 Merge Join  (cost=8.94..3748.82 rows=145860 width=192) (actual time=0.053..17.403 rows=50000 loops=1)
   Merge Cond: (supplier.s_suppkey = table1.sorted)
   ->  Index Scan using suppkey_idx on supplier  (cost=0.29..492.41 rows=10000 width=139) (actual time=0.034..1.846 rows=10000 loops=1)
   ->  Index Scan using sorted_idx on table1  (cost=0.43..310044.43 rows=10000000 width=53) (actual time=0.014..7.676 rows=50006 loops=1)
 Planning Time: 0.156 ms
 Execution Time: 18.450 ms
(6개 행)


postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_suppkey=unsorted;
                                                                QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 Merge Join  (cost=0.88..4372.18 rows=56025 width=192) (actual time=0.054..261.355 rows=50164 loops=1)
   Merge Cond: (supplier.s_suppkey = table1.unsorted)
   ->  Index Scan using suppkey_idx on supplier  (cost=0.29..492.41 rows=10000 width=139) (actual time=0.005..2.239 rows=10000 loops=1)
   ->  Index Scan using unsorted_idx on table1  (cost=0.43..619951.15 rows=10000000 width=53) (actual time=0.018..248.699 rows=50168 loops=1)
 Planning Time: 0.183 ms
 Execution Time: 262.835 ms
(6개 행)
```

Exercise 5-1 Answer

- SELECT * FROM supplier, table1 WHERE s_nationkey=sorted;
- SELECT * FROM supplier, table1 WHERE s_nationkey=unsorted;
- Merge join (index)
- Execution time: (sorted ≈ unsorted) (since the cardinality of n_nationkey is small)

```
postgres=# SET enable_memoize = off;
SET
```

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_nationkey = sorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Merge Join  (cost=981.82..3229.75 rows=149556 width=192) (actual time=1.659..8.519 rows=50000 loops=1)
   Merge Cond: (table1.sorted = supplier.s_nationkey)
   ->  Index Scan using sorted_idx on table1  (cost=0.43..310075.65 rows=9999748 width=53) (actual time=0.021..0.044 rows=126 loops=1)
   ->  Sort  (cost=981.39..1006.39 rows=10000 width=139) (actual time=1.633..3.106 rows=49996 loops=1)
         Sort Key: supplier.s_nationkey
         Sort Method: quicksort  Memory: 2061kB
         ->  Seq Scan on supplier  (cost=0.00..317.00 rows=10000 width=139) (actual time=0.009..0.461 rows=10000 loops=1)
 Planning Time: 0.227 ms
 Execution Time: 9.417 ms
(9개 행)


postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_nationkey = unsorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Merge Join  (cost=1034.76..1785.19 rows=55463 width=192) (actual time=1.738..9.390 rows=47954 loops=1)
   Merge Cond: (table1.unsorted = supplier.s_nationkey)
   ->  Index Scan using unsorted_idx on table1  (cost=0.43..620100.51 rows=9999748 width=53) (actual time=0.017..0.866 rows=120 loops=1)
   ->  Sort  (cost=981.39..1006.39 rows=10000 width=139) (actual time=1.718..3.291 rows=47952 loops=1)
         Sort Key: supplier.s_nationkey
         Sort Method: quicksort  Memory: 2061kB
         ->  Seq Scan on supplier  (cost=0.00..317.00 rows=10000 width=139) (actual time=0.005..0.410 rows=10000 loops=1)
 Planning Time: 0.152 ms
 Execution Time: 10.240 ms
(9개 행)
```

Exercise 5-2 Answer

- SELECT * FROM supplier, table1 WHERE s_nationkey=sorted;
- SELECT * FROM supplier, table1 WHERE s_nationkey=unsorted;
- Hash Join
- Execution time: sorted ≈ unsorted

```
postgres=# INSERT INTO supplier(s_suppkey, s_nationkey) SELECT generate_series(10001, 100000), generate_series(10001, 100000);
INSERT 0 90000
```

```
postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_nationkey = sorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=474458.95..1343480.20 rows=11553155 width=194) (actual time=1590.898..7326.117 rows=9999995 loops=1)
   Hash Cond: (table1.sorted = supplier.s_nationkey)
   ->  Seq Scan on table1  (cost=0.00..203125.48 rows=9999748 width=53) (actual time=0.007..697.895 rows=10000000 loops=1)
   ->  Hash  (cost=144410.09..144410.09 rows=9999109 width=141) (actual time=1540.849..1540.850 rows=10000000 loops=1)
         Buckets: 65536  Batches: 256  Memory Usage: 2040kB
         ->  Seq Scan on supplier  (cost=0.00..144410.09 rows=9999109 width=141) (actual time=0.012..508.256 rows=10000000 loops=1)
 Planning Time: 0.190 ms
 Execution Time: 7473.751 ms
(8개 행)


postgres=# EXPLAIN ANALYZE SELECT * FROM supplier, table1 WHERE s_nationkey = unsorted;
                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 Hash Join  (cost=425776.33..1343478.60 rows=11553155 width=194) (actual time=2182.991..8702.882 rows=9997787 loops=1)
   Hash Cond: (supplier.s_nationkey = table1.unsorted)
   ->  Seq Scan on supplier  (cost=0.00..144410.09 rows=9999109 width=141) (actual time=0.368..525.784 rows=10000000 loops=1)
   ->  Hash  (cost=203125.48..203125.48 rows=9999748 width=53) (actual time=2178.612..2178.613 rows=10000000 loops=1)
         Buckets: 131072  Batches: 128  Memory Usage: 7607kB
         ->  Seq Scan on table1  (cost=0.00..203125.48 rows=9999748 width=53) (actual time=0.005..706.679 rows=10000000 loops=1)
 Planning Time: 0.151 ms
 Execution Time: 8850.436 ms
(8개 행)
```