Exercise 11
Date of issue: 16th May 2017 Deadline: 23rd May 2017

Hints and Advices

- Query plans can be viewed by prepending `EXPLAIN` to the query. You may use this command to assist your analyses.
- For your analyses, do not assume that the tuples of a table are stored sorted on disk.
- PostgreSQL uses a cost-based query optimizer.
- Please take the primary keys into your consideration.

1 Selection and Indexes

Given two SQL statements:

1. `SELECT * FROM city WHERE Population > 10000000;`

2. `SELECT AVG(Population) FROM city;`

Provide the PostgreSQL execution plan for these queries and explain how the queries are executed and the reason why the strategies are chosen.

Create the following index:

`CREATE INDEX city_pop ON city USING BTREE (Population);`

How does this statement affect the execution plans of both of the given queries? Explain your answer.

2 Join Strategies

Assume relations r and s with schemas $R(A, B)$ and $S(C, D, E)$ and the following information:

- r has 100’000 tuples, s has 300’000 tuples.
- The size of a tuple in r is 20 bytes, the size of a tuple in s is 30 bytes.
- The disk block size is 4096 bytes.
- A tuple must be stored entirely in a block.
- The average time to retrieve a block from a disk is 2ms.
- r and s have separate buffers.
- The buffers have a size of one disk block.
Calculate the time for the block fetches of $r \bowtie_{A=C} s$ for the following join algorithms. State the formula you used and consider all possibilities to choose outer and inner relation.

1. Block Nested Loop Join.

2. Indexed Nested Loop Join under the assumption that a primary $B^+$-tree index on attribute $A$ with 3 levels is available.

3. Nested Loop Join with MRU under the assumption that the buffer size of the inner relation is 100 blocks and that the smaller relation is the outer one.

4. Merge Sort Join under the assumption that $r$ is sorted on $A$, $s$ is sorted on $C$, and all tuples with equal join values are on the same block.

5. Hash Join.

3 Optimization

1. Given the following query:

   ```sql
   SELECT Continent
   FROM encompasses, country
   WHERE country.Code = encompasses.Country;
   ```

   • Provide the PostgreSQL execution plan for this query and explain how the query is executed and the reason why this strategy is chosen.
   • Disable the hash join with `set enable_hashjoin=off`. Give the new query plan and explain it, explain why it is slower than the previous plan.
   • Disable the merge join with `set enable_mergejoin=off`. Give the new query plan and explain it, explain why it is slower than the previous plan.
   • Suggest a better strategy of the query execution, under the assumption that the attribute `Country` in relation `encompasses` is a foreign key, that references `country(Code)`.

2. Given the following query:

   ```sql
   SELECT DISTINCT Code
   FROM country;
   ```

   • Provide the PostgreSQL execution plan for this query and explain how the query is executed.
   • Suggest a better strategy of the query execution.