Ex4 - Join Algorithms

Due date: Monday December 17, 2018, 23:55

Submission instructions:

In this exercise you should create a single zip file named ex4.zip. Submit the zip file via the ex4 submission link on the course homepage from one user only (even if you are working in pairs). You should write both names, ids and user names in the pdf file. The zip file should contain the following 2 files:

- ex4.pdf: with your written parts of your answers to the questions below.
- ex4.sql: with the code for the practical part in Question 4.

Note: You should write your calculations in details, including verbal explanation and calculation steps. Explain your calculations as much as possible so it will be clear what you did.

Question 1

We want to compute the expression $R(A,B) \bowtie S(B,C)$.

1. Assume that B(R)=100,000, B(S)=10,000 and the buffer size is $M = 102$. There are no indexes, and B is a key in S.

   For each one of the algorithms below, calculate the cost of processing the expression. If the algorithm is not applicable explain why.

   (a) Block Nested Loops Join

   (b) Sort Merge Join

   (c) Hash Join

2. Assume now that the buffer size is $M = 1,002$, and again for each one of the algorithms above, calculate the cost of processing the expression. If the algorithm is not applicable explain why.

3. For each of the above algorithms, what is the minimal buffer size that would allow the algorithm to be applicable.

Question 2

We want to compute the expression

$$\sigma_{A>3 \wedge D=15}(\pi_{A,D}(R(A,B,C) \bowtie S(B,C,D)))$$

Assume that:

- The buffer size is $M = 502$.
- B(R)=60,000.
- B(S)=5,000, but due to updates and deletions, the blocks of table S are only 80% full on disk.
- V(R,A) = 120, V(R,B) = 100, V(R,C) = 500.
- V(S,B) = 2,000, V(S,C) = 100, V(S,D) = 8.
- Every attribute takes 20 bytes.
- Each block contains 3,000 bytes.

Projection is performed without duplication removal and there are no indexes available (and none can be built).

1. (a) What is the number of tuples in R?
   (b) What is the number of tuples in S?
2. What will be the size of the result:
   (a) in tuples?
   (b) in blocks?
3. What will be the optimal way to calculate the expression? Explain and draw a query plan.
4. What will be the I/O cost of the optimal calculation?

Question 3

We want to compute the following query:

$$\sigma_{C=6}(R(A,B) \bowtie S(B,C))$$

Assume that:

- $B(R) = 100,000$
- $B(S) = 10,000$

- the buffer size $M = 22$
- Attributes A and C each require 10 bytes, and attribute B requires 20 bytes. A pointer requires 4 bytes.
- a block contains 1000 bytes
- V(R,A)=10, V(R,B)=30, V(S,C)=100 and B is a key in S.

Currently, there are no indices on R or on S, however, *one* index may be built on S.

1. If we choose to not build any indexes, what would be the cost of computing the query using block nested loops join?
2. If we choose to build an index on S.B:
   (a) What will be the order of the index?
   (b) What will be the depth of the index?
   (c) What will be the cost of calculating the query using index nested loops join?
3. If we choose to build an index on S.C:
   (a) What will be the order of the index?
   (b) What will be the depth of the index?
   (c) What will be the cost of computing the query $\sigma_{C=6}S(B,C)$ using the index?
   (d) Can this index be used to speed up the calculation of complete query (appearing above) using block nested loops join? Explain your answer.
4. Draw the query plan of the optimal way to calculate the query, among those considered in this question.

Question 4

Read the following pages of the Postgres documentation:

- `https://www.postgresql.org/docs/9.6/static/sql-createindex.html` describing how to create an index
- `https://www.postgresql.org/docs/9.6/static/sql-explain.html` describing how to use the command ”explain”, which shows the execution plan of a statement.

The purpose of this question is to explore how a query plan may change, given the presence of an index. Write code that:

1. creates two tables
2. inserts data into the tables
3. uses explain to show the execution plan of a query of your choice that joins the tables (and may also perform additional operations)
4. adds an index to one or more of the tables
5. uses explain to show the execution plan of the same query as in 3. (You should create your query and indexes such that the results of the two explains are different.)
6. drops all tables and indexes created.

Your code should run properly in postgres!

You should submit the code, the output of the two explain commands, as well as your own explanation of the differences between the two query plans.