W4111 – Introduction to Databases
Sections 002, V002; spring 2022

Homework 4a – Written Assignment

Instructions

- The homework submission date/time is 2022-MAY-01 at 11:59 PM.
- Submission format is a PDF version of this document with your answers. Place your answers in the document after the questions.
- The name of your PDF must be <UNI>_S22_W4111_HW4a_Written.pdf. For example, mine would be dff9_S22_W4111_HW3a_Written.pdf
- You must use the Gradescope functions to mark the location of your questions/answers in the submitted PDF. Failure to mark pages will cause point deductions. **Please, please read the countless Ed posts, TA produced instructions and videos, etc. to prepare your submission.**
- You can use online sources but you must cite your sources. You may not cut and paste text.
- Questions typically require less than five sentences for an answer. You will lose points if your answer runs on and wanders.

  "Verbosity wastes a portion of the reader's or listener's life."

Questions

Question 1: Explain why a sparse index must also be a clustering index.

Answer:

For clustering index, data file is ordered on a non-key field. For sparse index, index records are not created for every search key, which means that the index is not based on keys. It's based on a non-key field. So a sparse index must be a clustering index.

Question 2: Briefly explain sparse, multi-level indexes and their benefits. Why can the other index be sparse?

Answer:

For sparse indexes, unlike dense index, index records are not created for every search key. An index record here contains a search key and an actual pointer to the data on the disk. So it needs less space, less maintenance overhead for insertion, and deletions.

Multilevel indexes compose of several layers. The outer layer only stores search-key values and data pointers. Through this way, more index record could be kept in the memory.

Multilevel indexes could be sparse because not all records are created for every search key.

Question 3: Indexes can significantly improve performance? What are some disadvantages of having many indexes?

Answer:
Indexing can help efficiently retrieve records from the database files based on some attributes on which the indexing has been done.

Disadvantage of indexing could be additional disk space for non-clustering indexes. The need for an update to the entire index when an Insert, Update, Delete is called on the table could also count as a disadvantage since this need slows performance.

Question 4: Briefly compare the pros and cons of B^+-Tree versus a Hash Index.

Answer:
B+ Tree is good for integer and date columns, where value comparison frequently goes.
B+ tree is more memory efficient than hash index.
B+ tree could do range search efficiently.
Hash Index is great for exact equality operations and runs much faster than B+ tree.

Question 5: Briefly explain the concepts of covering index/covered query.

Answer:
A covering index is a non-clustered index.
It includes all columns referenced in the covered query and therefore, the optimizer does not have to perform an additional lookup to the table in order to retrieve the data requested.

Question 6: Briefly explain the three main steps/stages in query processing.

Answer:

First stage is Parser/Translator, which verifies syntax correctness and generates a parse tree, then the tree is converted to logical plan tree that defines how to execute the query.

Second stage is Optimizer, which modifies the logical plan to define an improved execution. It also does query rewrite/transformation and determines how to choose among multiple implementations of operators.

Third stage is Engine, which executes the plan. In this stage, the plan may be optimized for execution.

Question 8: Explain the role of equivalent queries in query optimization.

Answer:
The goal of query optimization is to reduce the system resources required to fulfill a query, and ultimately provide the user with the correct result set faster. If equivalent queries are provided, then query optimization process will be much easier to execute.

Question 9: Assume that the ON clause in a JOIN on tables A and B compares columns a1 with b1 and column a2 with b2. What two properties must the ON condition have for the optimizer to be able to use a Hash Join?

Answer:

First property is that a relatively large amount of data is joined, or a large fraction of a small table is joined.
Second property is that the join is an equijoin.

Question 10: What is index selectivity? How does it factor into query optimization for JOINs?

Answer:
Index selectivity is how tightly a database index helps narrow the search for specific values in a table.
When doing query optimization for join, higher index selectivity would help the query optimization go faster and easier to execute