Database Systems WS 2019/20
Prof. Dr.-Ing. Sebastian Michel
M.Sc. Nico Schäfer / M.Sc. Angjela Davitkova
Exercise 8: Handout 17.12.2019, Due 06.01.2020 16:00 MEZ

DBS LAB TU KAISERSLAUTERN
https://dbis.cs.uni-kl.de

 [1]

We wish you happy holidays and a wonderful new year.

Question 1: Processing k-NN Queries in M-Trees (1 P.)

K Nearest Neighbors (k-NN) query $Q = (q, k)$ returns the k closest points (points that have the shortest distance) to the query point q.

a) Provide a pseudo code for k-NN query in an M-Tree.

Required submission: Pseudo code

b) Did you apply or can you think of pruning techniques that could be applied to reduce the number of points examined as candidates for the result? Describe why the conditions hold.

Required submission: Explanation of applied pruning techniques

Question 2: Extendible Hashing (1 P.)

Implement extendible hashing in a language of your choice. A basic Java template is available in OLAT.

Your program should take a bucket capacity k and a data file as input. Each line of the file should then be hashed and distributed, using extendible hashing, into buckets with a capacity of k.

Build the directory using the prefixes of the hash values. I.e.: Use the *first* d digits of the hash values. The buckets are empty in the beginning, thereby, the directory has size $d = 0$.

Your implementation must have the following features:

- Take data file as parameter
  - Calculate the hash of each line
- Take maximum bucket size as parameter
- After each insertion step, return the current directory (The buckets, their content, the local depth, the hash prefixes pointing to them, and the global depth)

The template already implements reading the file and hashing each line.

Required submission: Source code; Output after executing the program with parameters $k = 3$ and the data file in OLAT.

[1] Source: openclipart.org

Question 3: Linear Hashing (1 P.)

Perform linear hashing for the following given parameters:

Using the following sequence of hash functions:

$$H_i(K) = K \mod (2 \cdot 2^i) \text{ with } i \in \{0, 1, 2, \ldots, n\}$$

The hash table should be initialized with 2 buckets. Each bucket has a capacity of 3 entries. If more than $\beta > \frac{2}{3}$ of the table is occupied, controlled splitting should be performed.

Insert the following values in the given order:

$$27, 13, 28, 3, 21, 27, 8, 16, 28$$

Write down what happens during each insert. Also visualize your buckets after every split.

Required submission: Explanation of each insert step; Visualization of buckets after every split.

Question 4: Top-k Algorithms (1 P.)

Apply the FA and TA algorithm for $k = 2$, using addition as aggregation function, on the following three index lists. Write down all index list accesses, as well as the current top-k documents after each step. How many sequential and how many random accesses were executed?

L_1
d_1 0.8
d_3 0.7
d_4 0.4
d_2 0.4
d_8 0.4
d_6 0.3
d_5 0.1

L_2
d_2 0.9
d_4 0.8
d_3 0.5
d_1 0.4
d_5 0.3
d_7 0.2
d_8 0.1

L_3
d_3 0.9
d_6 0.8
d_4 0.7
d_1 0.3
d_5 0.2
d_7 0.2
d_8 0.2

Required submission: Explanation of each step in the algorithms; Top-k elements; Sequential/Random read accesses

Question 5: Potpourri (0 P.)

This is an optional exercise, don't submit anything. There will be no points given for answering these questions.

Below a list of review question that you can also expect to be given in the written or oral exams. You can use these questions to deepen your knowledge of the previous topics.

1. Give an example page reference string of length at least 4 where FIFO has less misses than LRU, if possible.
2. Explain, in your own words, the five minute rule.
3. Given a secondary index on an attribute that has one distinct value for each tuple. Is this index a dense or a sparse index?
4. In which cases do database systems sort data? Name at least 2.
5. Why does a B+ tree have a higher fan out than a B tree and what are the consequences?
6. Explain one advantage and one disadvantage of a heap-organized file compared to a sequentially ordered file.
7. What kind of queries usually benefit most from having a clustered index?
8. Does the order of the attributes of a composite index influence the efficiency of a query over these attributes?
9. Describe how to remove duplicates by hashing.
10. Explain why it sometimes can be better to sort the entire relation instead of accessing a large chunk of sorted data using a non-clustered index.
11. When could it make sense to prefer a schema in 3NF to a schema in BCNF?
12. Is it possible that a nested loop join is more efficient than a sort merge join?
13. How many left-deep trees are possible for star queries with n relations if no cross products are allowed?
14. Is a hash join always better than a nested loop join?
15. Explain the difference between biased and unbiased estimator.
16. Describe the histogram estimated using the following function: $f(x) = n$.
17. Given a random variable X with distribution function F. What is the p quantile of X?
18. Which initial values are described by the wavelet transform $[4, 2, 1, 0]$?
19. It is usually assumed that the individual predicates of a conjunctive predicate are independent, when computing the overall selectivity. Given a real-world example where this is far off and explain what the consequences can be for the query optimizer.
20. Explain which changes have to be performed on an materialized view, calculating an aggregation, with functions COUNT, SUM, and AVG when adding or removing tuples from the base relation.