Ceng352- Written Assignment 2 Sample Answers

April 2020

Q1

a The query only checks (age, grade) couple, since our index is on (age, grade) couple we can use the index file without checking the actual file. The index only plan is applicable to this query.

b In this query, we have 'gender=Female' part in the where clause which can not be found using the index only plan. For this query, it is not applicable.

Q2

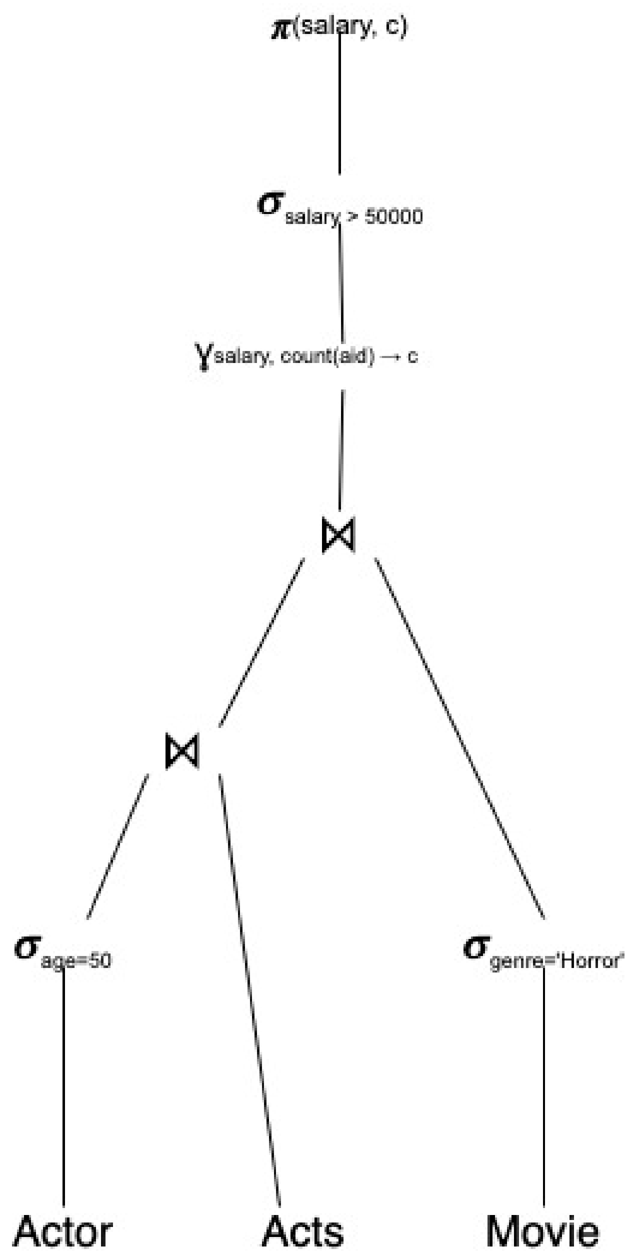
a Hash index, since there is equality check in the query.

b Since this is a range query and probably (assuming values of R are distributed uniformly) this query will return almost 20000 values, using clustered B+ tree will be the best option.

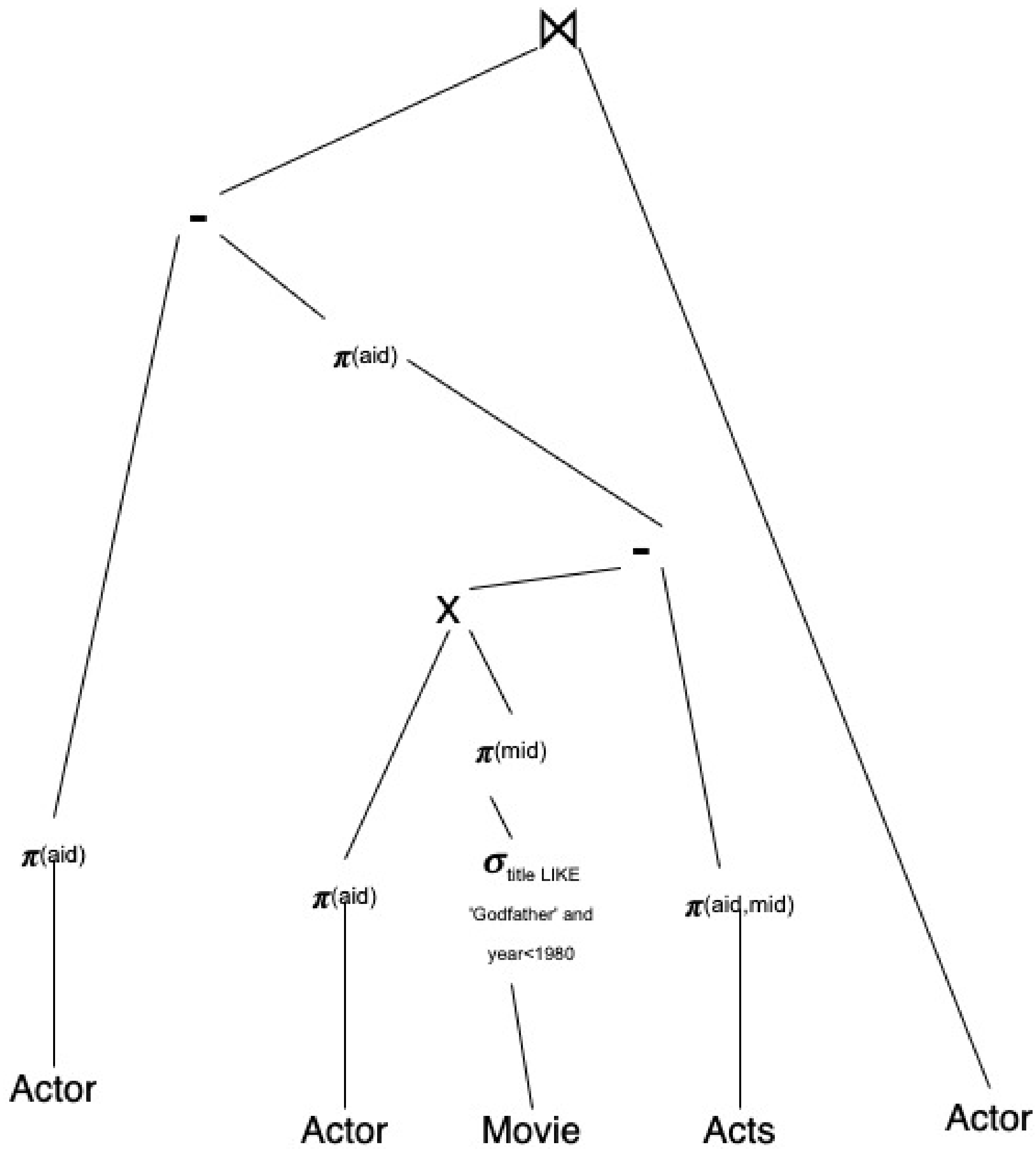
c Since this is again a range query, B+ tree will be the best option. However, again assuming R values are distributed uniformly, the query will return like 8-10 values, using clustered/unclustered B+tree will give similar results.

d Will return all the records except 1 (assuming uniform values). Thus, searching through the heap file will give the best solution since we will get (and check) almost all values.

Q3 a The resulting plan will be:



b Note that this is a division operation and the resulting plan will be:



Q4

a Cost of block nested loop join assuming R is the outer relation will be:
B(R) + $\frac{B(R)B(S)}{(M-2)}$

= 2000 + 2000*500/40 = 2000 + 25000 = 27000 I/O

b Cost of block nested loop join assuming S is the outer relation will be: B(S) + $\frac{B(S)B(R)}{(M-2)}$

= 500 + 500*2000/40 = 500 + 25000 = 25500 I/O

c As general, we can not apply one-pass algorithm (R and S are too large for that operation). However, in this example we can not apply two-pass algorithm also, since new runs of R created by sorting will not fit in the memory also (B(R) > M^2). We need to make it more than two-pass.
First, sorting S will give 500/42 = 11.9 → 12 runs. These runs will fit in the memory we don't need a second pass for that. The cost of this operation is 2B(S) (read+write).
Sorting R will give 2000/42 = 47.6 → 48 runs. This operation will cost 2B(R).
We need to merge these runs into bigger least number of runs. Again we will sort R as we did before into 2 runs(24/24). This operation will cost 2B(R).
Now we have runs of R and S which can fit in the memory separately or together. We need to merge them. The cost of the merge will be B(R) + B(S).
By adding all of them together we will get → 2B(S) + 2B(R) + 2B(R) + B(R) + B(S) = 5B(R) + 3B(S) = 2000*5 + 500 *3 = 11500.

d We will use our first hash function in order to create partitions of R:
Read R, use the hash function, write partitions to the disk 2B(R).
Read S, use the hash function, write partitions to the disk 2B(S).
Read partitions from R by using the second hash function, hash them in the memory, read partitions from S by using the second hash function find matchings. The cost is B(R) + B(S).
We are able to apply these since min(B(R), B(S)) < M^2 → 500 < 422
The total cost will be 3B(R) + 3B(S) = 3*2000 + 3*500 = 7500

e For each tuple of R, we need to find corresponding match of it using the index on the S. Read R, for each tuple, check the index and find the matches.
For the calculations we will need V(S,b). Since b is the primary key of S all will be unique and V(S,b) = 5000.
If it is clustered the cost will be: B(R) + T(R)B(S)/V(S,b) = 2000 + 20000*500/5000 = 2000 + 2000 = 4000.
If it is unclustered the cost will be: B(R) + T(R)T(S)/V(S,b) = 2000 + 20000*5000/5000 = 2000 + 20000 = 22000.

Q5

a $\frac{(N*(m10*m11*m12)*(tswim))}{(m1*m2*...*m12)*(tswim*tmitten*tgarden)}$

and the bottom part is equal to N^2

The result will be $\frac{(m10*m11*m12)*tswim}{N}$

b We've assumed that the data is uniform. However, in real life, this will not be the case. For example, in the m6 the sales of the swimsuits will be larger than m1. These can cause problems in this estimation. Specific to this question, people generally will not buy swimsuits in the 10^{th}, 11^{th} and 12^{th} months of the year, our estimations will probably be higher than actual values.