Homework 3 (70 Points)

As for the previous homework, submit one .zip file containing five text files named Q1.txt to Q5.txt, each containing the answer to the corresponding question. For questions 1, 2, 3, and 5, the first line of the file contains the raw number of page I/Os (without units) and the following paragraph a justification. For question 4, the first line contains either “yes” or “no” (without quotes) and the following paragraph contains a justification.

For the following exercises, use the same cost model, assumptions, and conventions as in the lecture (in particular: slide deck 13 (joins) and 15 (query plan costing and query optimization)).

For the following four questions, we make the following assumptions:

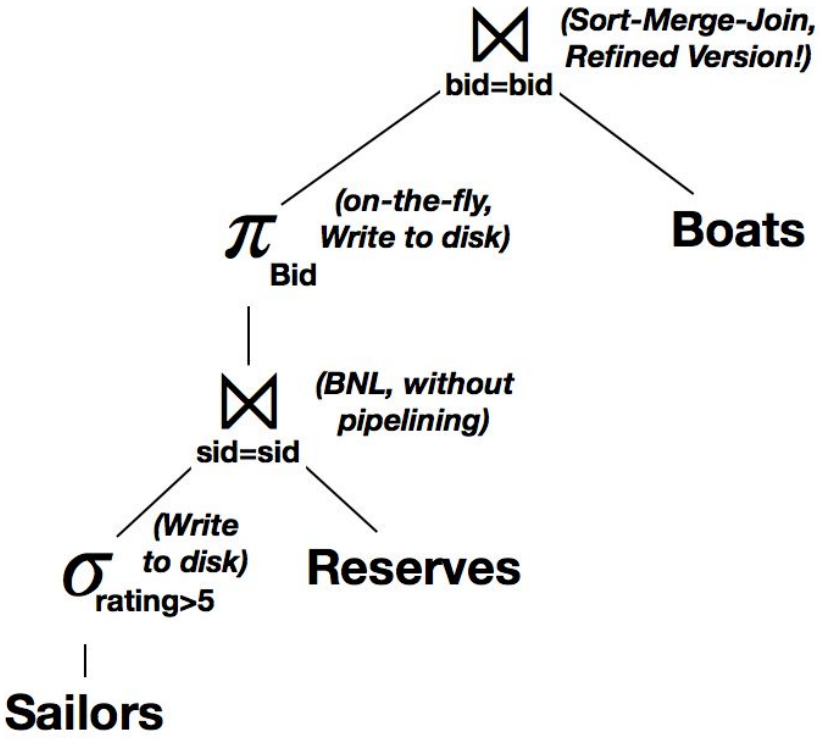
- We join two tables, R and S, via an equality join condition
- Each tuple in R consumes 80 bytes and R contains 10,000 tuples
- Each tuple in S consumes 100 bytes and S contains 50,000 tuples
- We consider disk pages of 8,000 bytes
- We have 12 buffer pages in main memory

Q1) (10 Points) What is the cost of a Block Nested Loops join with R as outer relation (assuming that the join exploits all buffer pages)?

Q2) (10 Points) We assume that an unclustered hash index is available on the join column for S. The index stores data entries of type “Alternative 2”. What is the cost of an Index Nested Loops join with R as outer relation? Assume the cost of retrieving the index data entries for a given search key (not including cost of retrieving actual data!) is in average 1.2 I/Os.

Q3) (10 Points) What is the cost for executing a sort-merge join between R and S? Calculate cost for the simple sort-merge join version (not the refined version!) that is used on slide 17 from slide deck 13 (joins).

Q4) (10 Points) Do we have enough main memory to execute a hash join between R and S with a single partitioning pass? Justify by explaining why main memory is sufficient for both phases (partitioning phase and join phase). Assume that each partition has the same size.



Q5) (30 Points) Calculate the processing cost for the query plan shown above, we make the following assumptions:

- Table Sailors contains 100,000 sailors with columns sname (20 bytes per tuple), sid (4 bytes per tuple), and rating (4 bytes per tuple).
- Table Reserves contains 200,000 reservations with columns sid (4 bytes per tuple), bid (4 bytes per tuple), and date (8 bytes per tuple).
- Table Boats contains 10,000 boats with columns bid (4 bytes per tuple) and name (20 bytes per tuple).
- Each reservation is linked to exactly one boat and sailor (via the bid and sid columns).
- Sailor ratings are values between 1 and 10 (both inclusive).
- Reservations are uniformly distributed over boats and sailors (e.g., the number of reservations is approximately the same for all sailors) and sailors are uniformly distributed over ratings (i.e., the number of sailors for each rating is the same).
- We have 20 buffer pages, each page stores 8,000 bytes.