CS 623

PROJECT

Guidelines

- This is a group project that you will have to do in a group of 3 students (maximum).
- Post your team group as well as the data source for your group's data set in the spreadsheet.
- You will use PostgreSQL (rather than MySQL).
- Your code should also be on your individual GitHub. This is where I will check it. The code is developed as a team but available on the GitHub of participating students.
- You have two parts, the Practical and the Theory part. There is an extra 1 mark available for attempting the project.

Deliverables

- Code in GitHub(individually) and link to the github. I will check the code there.
- Submit a Video of < 3 minutes to show and explain your work
- Screenshots of the code plus output.
- PDF/Word doc of solutions to theory questions

Description

Involves working with spatial data and utilizing the access methods and query executions and optimizations we would discuss in class. The project would involve writing SQL queries to retrieve information such as the locations of specific features, distances between points, and areas of interest. Using indexing, aggregate and join executors, sort+ limit executors, sorting, and top-N optimization.

Practical Part (75%)
Goal

Creating a Geographic Information System (GIS) Analysis: A project that involves analyzing geographic data such as maps and spatial data. You will need a database that supports spatial data types, like PostgreSQL (PostGIS).

1. **Retrieve Locations of specific features (10 marks)**
2. **Calculate Distance between points (10 marks)**
3. **Calculate Areas of Interest (specific to each group) (10 marks)**
4. **Analyze the queries (10 marks)**
5. **Sorting and Limit Executions (10 marks)**
6. **Optimize the queries to speed up execution time (10 marks)**
7. **N-Optimization of queries (5 marks)**
8. **Presentation and Posting to Individual GitHub (5 marks)**
9. **Code functionality, documentation and proper output provided (5marks)**

Each member of the team posts the code of the project in GitHub. (INDIVIDUAL)

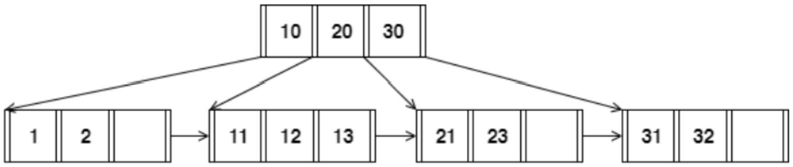
THEORY PART (24%)

You have 12 Theory questions, each with 2 marks.

1.) We have a file with a million pages (N = 1,000,000 pages), and we want to sort it using external merge sort. Assume the simplest algorithm, that is, no double buffering, no blocked I/O, and quicksort for in-memory sorting. Let B denote the number of buffers.

How many passes are needed to sort the file with N = 1,000,000 pages with 6 buffers?

2.) Consider the following B+tree.



When answering the following question, be sure to follow the procedures described in class and in your textbook. You can make the following assumptions:

• A left pointer in an internal node guide towards keys < than its corresponding key, while a right pointer guides towards keys ≥.

• A leaf node underflows when the number of keys goes below [(d−1)/ 2] e.

• An internal node(root node) underflows when the number of pointers goes below d /2 .

How many pointers (parent-to-child and sibling-to-sibling) do you chase to find all keys between 9 ∗ and 19∗ ?

3.) Answer the following questions for the hash table of Figure 2. Assume that a bucket split occurs whenever an overflow page is created. h0(x) takes the rightmost 2 bits of key x as the hash value, and h1(x) takes the rightmost 3 bits of key x as the hash value

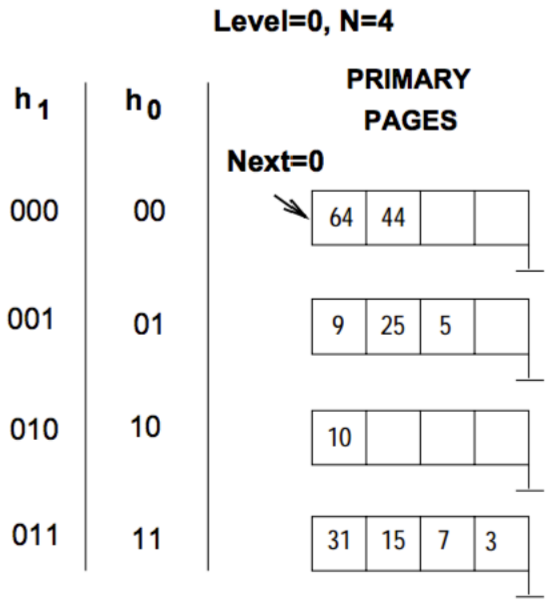


Figure 2: Linear Hashing

What is the largest key less than 25 whose insertion will cause a split?

4.) Consider a sparse B+ tree of order d = 2 containing the keys 1 through 20 inclusive. How many nodes does the B+ tree have?

5.) Consider the schema R(a,b), S(b,c), T(b,d), U(b,e).

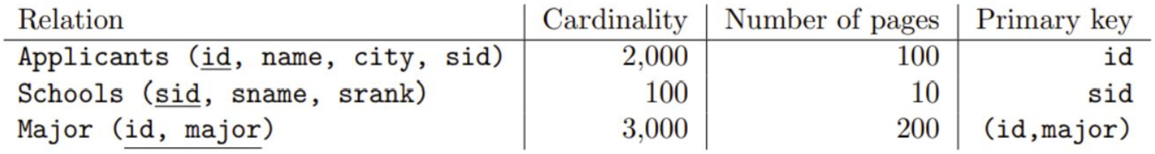
Below is an SQL query on the schema:
SELECT R.a
FROM R, S,
WHERE R.b = S.b AND S.b = U.b AND U.e = 6
For the following SQL query, I have given two equivalent logical plans in relational algebra such that one is likely to be more efficient than the other:
I. $\pi a(\sigma c=3(R \bowtie b=b (S)))$
II. $\pi a(R \bowtie b=b\ \sigma c=3(S)))$

Which plan is more efficient than the other?

6.) In the vectorized processing model, each operator that receives input from multiple children requires multi-threaded execution to generate the Next() output tuples from each child. True or False? Explain your reason.

7.) How can you optimize a Hash join algorithm?

8.) Consider the following SQL query that finds all applicants who want to major in CSE, live in Seattle, and go to a school ranked better than 10 (i.e., rank < 10).



Relation	Cardinality	Number of pages	Primary key
`Applicants (id, name, city, sid)`	2,000	100	`id`
`Schools (sid, sname, srank)`	100	10	`sid`
`Major (id, major)`	3,000	200	`(id,major)`

SELECT A.name

FROM Applicants A, Schools S, Major M

WHERE A.sid = S.sid AND A.id = M.id AND A.city = 'Seattle' AND S.rank < 10 AND M.major = 'CSE'

Assuming:

- Each school has a unique rank number (srank value) between 1 and 100.

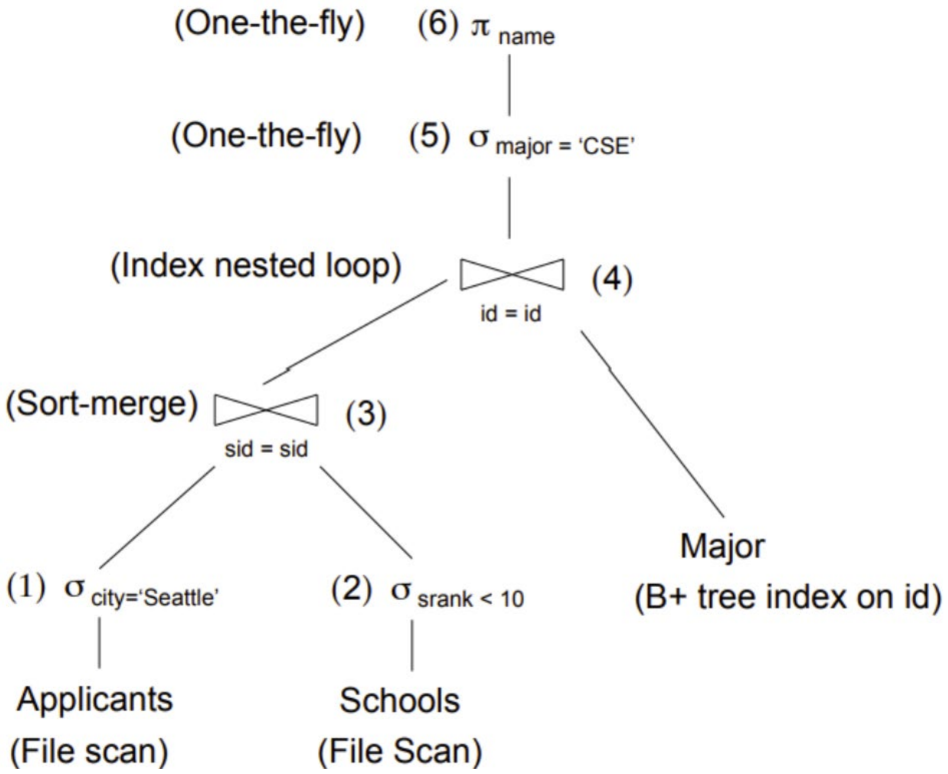
- There are 20 different cities.

- Applicants.sid is a foreign key that references Schools.sid.

- Major.id is a foreign key that references Applicants.id.

- There is an unclustered, secondary B+ tree index on Major.id and all index pages are in memory.

You as an analyst devise the following query plan for this problem above:



What is the cost of the query plan below? Count only the number of page I/Os.

9.) Consider relations R(a, b) and S(a, c, d) to be joined on the common attribute a. Assume that there are no indexes available on the tables to speed up the join algorithms. • There are B = 75 pages in the buffer

• Table R spans M = 2,400 pages with 80 tuples per page

• Table S spans N = 1,200 pages with 100 tuples per page

Answer the following question on computing the I/O costs for the joins. You can assume the simplest cost model where pages are read and written one at a time. You can also assume that you will need one buffer block to hold the evolving output block and one input block to hold the current input block of the inner relation.

A.) Assume that the tables do not fit in main memory and that a high cardinality of distinct values hash to the same bucket using your hash function h1. What approach will work best to rectify this?

B.) I/O cost of a Block nested loop join with R as the outer relation and S as the inner relation

10.) Given a full binary tree with 2n internal nodes, how many leaf nodes does it have?

11.) Consider the following cuckoo hashing schema below:

Both tables have a size of 4.The hashing function of the first table returns the fourth and third least significant bits: h1(x) = (x >> 2) & 0b11.The hashing function of the second table returns the least significant two bits: h2(x) = x & 0b11.

When inserting, try table 1 first. When replacement is necessary, first select an element in the second table. The original entries in the table are shown in the figure below.

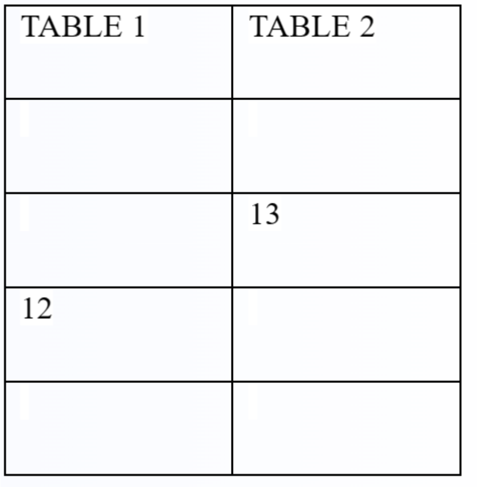


TABLE 1	TABLE 2
	13
12	

What sequence will the above sequence produce? Choose the appropriate option below:

a.) Insert 12, Insert 13
b.) Insert 13, Insert 12
c.) None of the above. You cannot have more than 1 Hash table in Cuckoo hashing
d.) I don't know