Assignment 5

Query Processing & Optimization

Assigned: Oct 28, 2024
Due: Nov 11, 2024. 11:59:59PM
Total Points: 50

Objectives

The purpose of this assignment is to go through some exercise of basic query processing, indexing strategies, and query optimization. Specifically, this assignment will help you practice your knowledge on the following points:

- Basic knowledge of DBMS query processing pipeline.
- How to build indexes for tables in DBMS.
- How to choose indexes for specific workloads.
- How to do basic push down optimization for a logical plan.
- How to do simple join cost estimation.

Please submit your solution to GradeScope before due date above. If you have further questions, please contact the instructor or any TA.

Part I: Query Processing & Indexing [30pts]

1. [10pts] Describe **a concrete case** where answering a database query with an index is **strictly slower** than scanning through the whole file. In particular, please describe the properties of data, query, and index in such case.

Your answer should include all important details, including the table schema, its cardinality, what indexes (type and attributes) are built, what query is being asked, what properties the table data and query result should have.

2. [20pts] Consider a relation with the schema "Student(SID, firstName, lastName, GPA, program)" and we have several query workloads running upon it.

You can assume the following fact of this table:
(a) this table has more than a million records;
(b) "(firstName, lastName)" is a candidate key of the table;
(c) students are evenly distributed across 20 programs;
(d) GPA of all students form a normal distributions around 3.0 with a standard deviation of 1.0.
Answer the following questions:

Notes: *You should be explicit of index type you are using when writing CREATE INDEX statements.*

(1) **[4pts]** Write a SQL statement to build an index to facilitate fast point lookup for students' last names.

(2) **[4pts]** Write a SQL statement to build an index to facilitate fast range search for students' GPA.

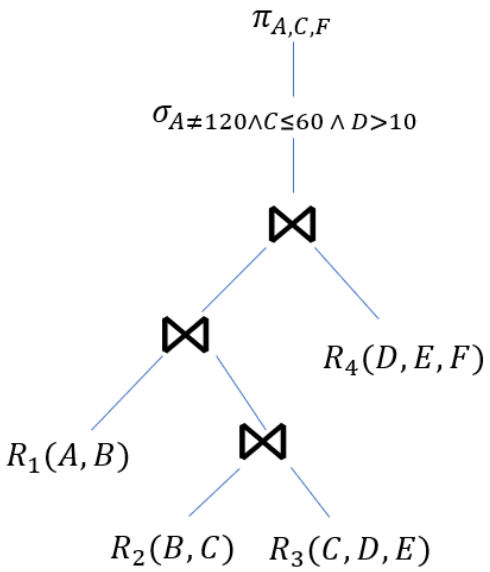
(3) **[5pts]** Suppose you have a B-Tree index built on (program, GPA) attribute, what queries over the students table will be benefited from it. Please list all of them including query types (point lookup or range query) and query attributes.

(4) **[7pts]** Suppose you have a workload having equal amount of queries asking for: (a) students in a specific program; (b) students whose GPA are in a small given range ($\pm$ 0.1); (c) students has a specific full name.

Right now, we want to accelerate this workload, but you are only allowed to build **two** indexes over the student table. Explain what type and on which columns you want to build index on and briefly justify your choice.

Part II: Query Optimization [20pts]

1. [8pts] Consider the logical plan below. Conduct push down optimization for both filter and projection as much as possible.



Hint: You may want to add additional operators so as to push down some of the original operator further while maintaining logical equivalence.

2. [12pts] Consider the following join below. Currently, we know the cardinality of table $R1$, $R2$ and $R3$ are 50000, 30000, 10000 respectively. The size of intermediate join results for $R1 \bowtie R2$, $R2 \bowtie R3$, and $R1 \bowtie R3$ are 10000, 1000, and 4000. The final join result has 500 records.

SELECT $*$ **FROM** R1 **NATURAL JOIN** R2 **NATURAL JOIN** R3;

Suppose we use BNLJ for all the joins involved, each page can hold 200 records, and we have a buffer size of 12 pages. Answer the following questions:

(1) **[6pts]** If we first join $R1$ and $R3$, then join its output with $R2$, what is the I/O cost executing this query?

(2) **[6pts]** Explore all the possible join orders and choose the best plan. You should put down detailed calculations for the cost of different physical plans.