CAS CS 660: Grad. Intro to Database Systems
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS660/

CAS CS 660: Grad Intro to Database Systems – Fall 2023 - Written Assignment #4

Due: 11/23 11:59 PM in Gradescope

Problem 1. [30 pts]

Consider the join $R \bowtie S$ where the join predicate is $R.a = S.b$, given the following metadata about R and S:

- Relation R contains 20,000 tuples and has 10 tuples per page (block)
- Relation S contains 5,000 tuples and has 10 tuples per page (block)
- Attribute b of relation S is the primary key for S, and every tuple in S matches 3 tuples in R
- There exists a unclustered B+-tree index on $R.a$ with height 3
- There exists a clustered B+-tree index on $S.b$ with height 2
- The main memory buffer holds 5

blocks

Answer the following questions:

1. If $R \bowtie S$ is evaluated with a block nested loop join, which relation should be the outer relation? Justify your answer. What is the cost of the join in number of I/O's?
2. If $R \bowtie S$ is evaluated with an index nested loop join, what will be the cost of the join in number of I/O's? Which index you will use? Show your cost analysis.
3. What is the cost of a plan that evaluates this query using sort-merge join. Show the details of your cost analysis.
4. Evaluate the cost of computing the $R \bowtie S$ using hash join assuming: i) The main memory buffer can hold 202 blocks, ii) The main memory buffer can hold 4 blocks.

Problem 2 [20 pts]

Consider the following relational schema:
Suppliers (sid: integer, sname: char(20), city: char(20)
Supply(sid: integer, pid: integer)
Parts(pid: integer, pname: char(20), price: real)

Suppose we are given the following **query:**

```
SELECT S.sname, P.pname
FROM Suppliers S, Parts P, Supply Y
WHERE S.sid = Y.sid AND Y.pid = P.pid AND
      S.city = 'Madison' AND P.price ≤ 1000
```

Briefly answer the following questions.

1. What information about these relations does the query optimizer need to select a good query execution plan for the given query?
2. What indexes might help process this query? Explain briefly about your choices.
3. How does adding DISTINCT to the SELECT clause affect the plans produced?

4. How does adding ORDER BY sname to the query affect the plans produced?
5. How does adding GROUP BY sname to the query affect the plans produced?

Problem 3 [50 pts]

Consider a database containing information about car accidents between 2000 and 2019, including the cars involved and their owners.The database has the following tables:

Car(*license, year, make, model*)
Accident(*license, accident_date, damage_amount, zipcode*)
- zipcode in Accident is the place where accident took place
- assume that the same car does not get into an accident twice in a day

Owner(*SSN, license, name, street, city, zipcode*)
- assume each owner has only one licensed car

The statistics and other catalog information for the database are as follows:

NTuples(Car) = 10,000
NTuples(Accident) = 20,000
NTuples(Owner) = 10,000
NPages(Car) = 100
NPages(Accident) = 1000
NPages(Owner) = 500
NDistinct(Car.make) = 50
NDistinct(Owner.City) = 20
Min(Accident.accident_date) = 01/01/2000
Max(Accident_date) = 12/31/2019.

Additionally:
- All indexes use Alternative 2 for the data entries.
- All BTrees have 100 keys per node, hash indexes have 100 keys per bucket,; assume BTrees are 3 levels deep, and the cost for a hash look up is 1.2 I/Os on average.
- An unclustered Linear hash index exists on Car(make)
- A clustered B+Tree index on Accident(accident_date)
- An unclustered Linear hash index exists on Owner(city)

Consider now the following query:
SELECT O.name, A.damage_amount FROM
Car C, Accident A, Owner O
WHERE C.license = A.license AND C.license = O.license AND C.make = "Ford" AND
A.accident_date < "01/01/2005" AND O.city="Boston";

Answer the following questions:

1. Provide a query tree (operation tree) that represents an evaluation of the SQL query above. Push the selection operations down as much as possible. No need to provide any algorithms for evaluating the operations here, just provide the tree.

2. Give the estimation of the results (how many tuples) after you apply the selection operations to Accident, Car, and Owner relations.

3. Assume that only Block Nested Loop Join (BNLJ) and Sort-Merge Join (SMJ) join algorithms are supported. Using the **System R query optimizer**, compute the best evaluation plan for this query and provide the estimated cost in number of I/ Os. Assume that the buffer has 7 pages.