Middle East Technical University

Department of Computer Engineering

CENG 352

Database Management Systems

Spring 2020–2021

Written Assignment 2

1 Query Processing

1.1 Part 1

a. Assume that we have a relation named Actor and its schema given as follows:

Actor(aid, name,age,net_worth)

Moreover a query on Actor relation is given below:

```
SELECT A.age, MAX (net_worth)
FROM Actor A
WHERE A.age <= 30
GROUP BY A.age
HAVING COUNT(*) > 2;
```

First explain what this query aims to find. Then, find minimal search key for B+ tree index that allows this query to be evaluated using index-only plan.

b. Assume the following relation is given:

Movie(mid, title, year, genre,budget)

Now answer which of the following queries can be evaluated with index-only plan if there is a composite index on (year, budget)

```
SELECT year, COUNT(distinct *)
FROM Movie
WHERE budget > 1000000
GROUP BY year;

SELECT year, AVG(budget)
```

```
FROM Movie
WHERE year > 2010 AND genre = 'Horror'
GROUP BY year;
```

1.2 Part 2

Given the following database schema:

```
Student(sid, sname, age, gender)
Course(cid, course_name)
Takes(sid, cid, semester, grade)
```

For each of the queries below, show an equivalent logical query plan.

```
SELECT T.grade, COUNT(T.sid)
FROM Student S, Takes T, Course C
WHERE S.sid = T.sid and C.cid = T.cid and C.cid = 'Ceng140' and gender ='F'
GROUP BY T.grade
HAVING T.grade > 3.00;

SELECT *
FROM Student S
WHERE NOT EXISTS (
    (SELECT C.cid
    FROM Course C
    WHERE C.course_name LIKE '%database%' AND C.credit >=3)

    EXCEPT

    (SELECT T.cid
    FROM Takes T
    WHERE T.sid = S.sid)
);
```

2 Join Algorithms

Consider R $\underset{R.x=S.y}{\bowtie}$ S given the following information about the relations R.x=S.y to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

- Relation R has 40000 tuples and 10 tuples per page.
- Relation S has 8000 tuples and 15 tuples per page.
- Attribute S.y is the primary key of S.
- Both relations are stored as simple heap files and neither relation has any indexes.
- 62 buffer pages are available in memory.

a. What is the cost of the join using **block nested loops join**, assuming R is the outer relation.

b. Answer part a, assuming S is the outer relation.

c. What is the cost of the join using **sort-merge join**. Explain page states in the memory (how they are used). Also, specify the disk inputs and outputs for each step.

d. What is the cost of the join using **hash join**. Assume that the hash function works perfectly. Similar to the part c explain memory state and disk IO.

e. Assume that we add index to the S on y. Describe how the join operation is affected and calculate the cost of the join using **index-nested loop join** for each case:

   i. S has clustered index on y.
   ii. S has unclustered index on y.

f. Now assume that relation S has another attribute z and V(S,z) = 20. We want to make selection on equality ($\sigma_{z=v}$), calculate the cost of this operation for the following:

   i. S has clustered index on z.
   ii. S has unclustered index on z.
   iii. S has no index on z.

   Then explain the reasons of the differences between the costs. Which one would you choose or which one is not giving good results.

3 Physical query plan

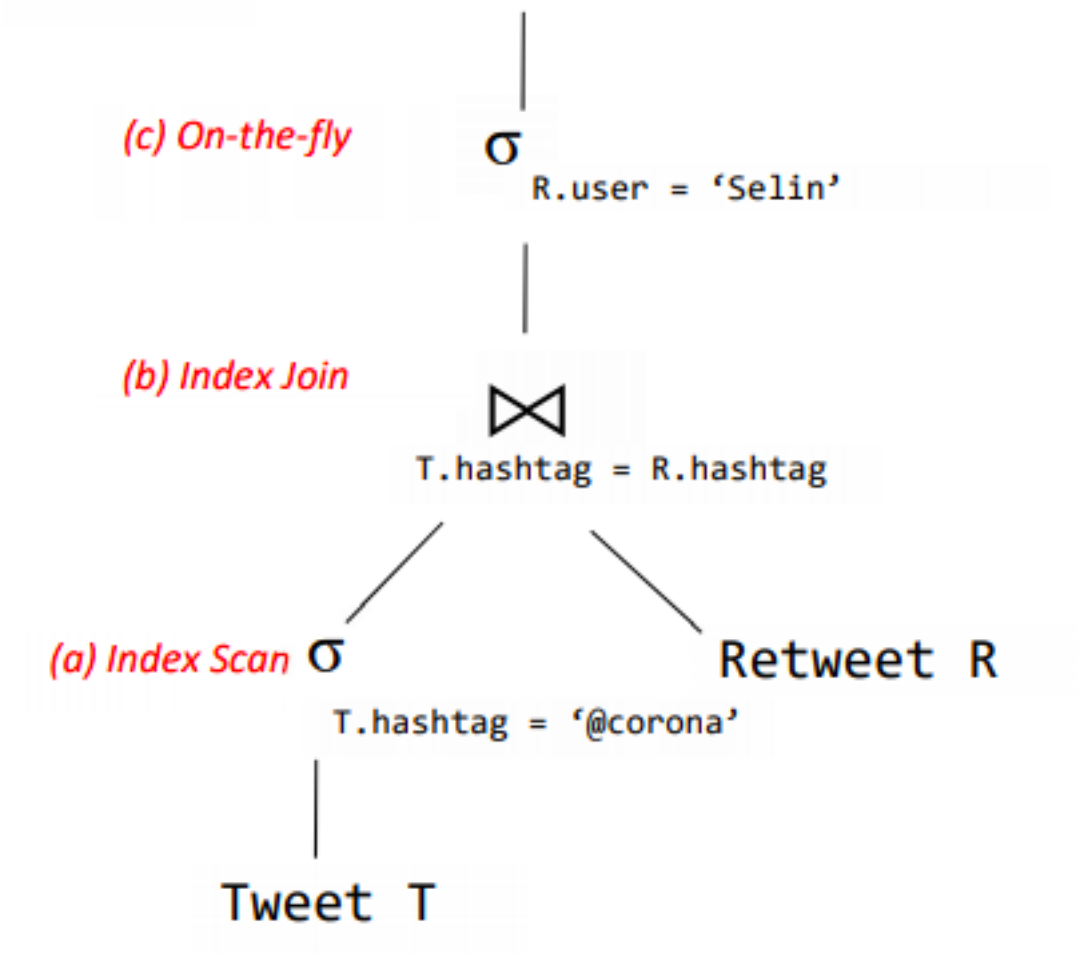
Consider the following database schema:
Tweet(author, hashtag)
Retweet(hashtag, user)
Note that there are no key nor foreign key constraints in the schema.
We are given the following statistics and indexes:

Tweet T	Retweet R
B(T) = 4000 pages	B(R) = 90,000 pages
T(T) = 600,000 records	T(R) = 3,000,000 records
V(T, author) = 2000	V(R, hashtag) = 30,000
V(T, hashtag) = 50,000	V(R, user) = 1000

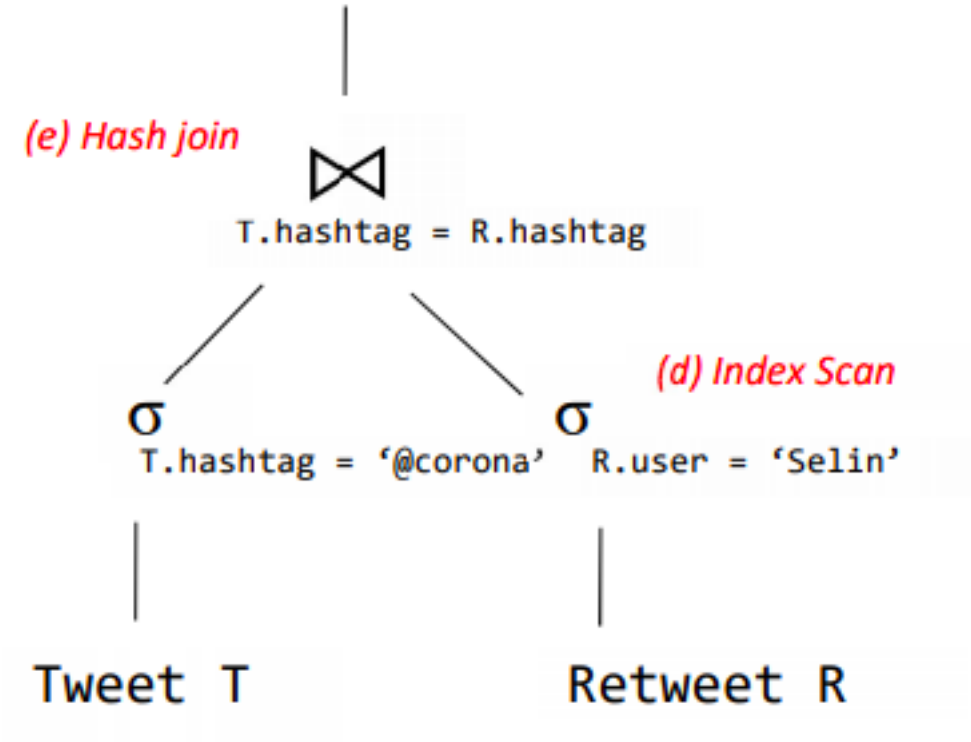
Indexes:

- There are clustered indexes on Tweet.author and Retweet.hashtag
- There are unclustered indexes on Tweet.hashtag and Retweet.user

Plan 1



Plan 2



The query optimizer considers two physical query plans as shown in the following figures. Assuming that we have unlimited memory and use pipelined execution, you are expected to calculate the estimated cost of each operation in the given plans. Your answer should include the number of disk I/Os and the result size of each operation. Assume that @corona is a value of Tweet.hashtag and Selin is a value of Retweet.user; that the values are evenly distributed.

Simple integer answers will not be sufficient. You should show your work.

a. Index scan:
   Cost = ?
   Result size = ?

b. Clustered index join:
   Cost = ?
   Result size = ?

c. On-the-fly selection
   Cost = ?
   Result size = ?

d. Index scan
   Cost = ?
   Result size = ?

e. In memory hash join, pipelined
   Cost = ?
   Result size = ?

f. What is the total IO cost of each plan? Which plan would you choose to minimize the total IO cost?

4 Experiments

For this part you should be able to perform some experiments using PostgreSQL. PostgreSQL supports Nested Loop Join, Hash Join and Merge Join. You will use Yelp Dataset that we've used in Project 1. If you have deleted the dataset you can download it again from here and create the same setup as Project 1. After you prepare the setup you should be able to do the following (Please read notes before starting):

a. Perform natural join on the tables Business and Tips and try to guess which algorithm will PostgreSQL choose and then show actual query execution plan. (You can send a screenshot of the result, you should find the way yourself).

b. For part a, try to create proper indexes on tables Business and Tips. Calculate execution times (using features of PostgreSQL) before and after creating the indexes. Explain the reason of difference between the execution times.

c. Perform another type of join where Business.business_id < Tip.business_id. Try to guess which algorithm will PostgreSQL choose and then show actual query execution plan (You don't need to find real execution time).

d. Now for part a and c, whichever method postgreSQL chooses try to disable it and again check which method postgreSQL chooses now (try each independently, i.e. don't disable both methods at the same time).

e. Now see the answer of question 7 given below:

```
Select B.state, AVG(B.stars) as avg_stars
From Business B
Group By B.state
Order By avg_stars DESC
LIMIT 10;
```

How can you reduce the execution time of that query without changing the query? Explain your answer. (You don't need to calculate actual execution time).

f. Do you think your changes on part e, affect other queries of Project 1? Explain why or why not.

g. All your answers until now should be written to pdf (screenshot or copy-paste is OK). Also, just like WHW1, dump your database and create an 'eXXXXXXX.sql' file, send it along with the pdf.

Note 1: For all the parts, add all the commands (you've used to find the answer) to the pdf.

Note 2: Your guesses about query plans (for part a and c) will not be graded, so please actually try to think about it and be honest with your guesses.

5 Submission

You should send a pdf file, named 'eXXXXXXX.pdf' (your seven-digit ID number) contains your answers. You can prepare the pdf using both 'DOCS' or 'Latex', doesn't matter. Moreover, you can send a scanned version of your handwritten answers, but please be sure that it is **readable**.

For Question 3, if you completed all of the steps you should have created an 'eXXXXXXX.sql' file (your seven-digit ID). Compress both '.pdf' file and '.sql'file to 'eXXXXXXX.zip' file ('.rar' or '.tar.gz' are also accepted) and upload it.