December 07, 2017
Arnab Chakrabarti, Rihan Hai

Implementation of Databases (WS 17/18)

Exercise 5

Due until December 19, 2017, 10am.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 5.1 (Query Optimization) (12 pts)

Consider a relation $Customer(ID, Name, Street, Birthday)$ containing 20,000,000 records, where each data page of the relation holds 40 records.

1. Suppose $Customer$ is organized as a sorted file with indexes, and $Customer$ is stored in $Customer.ID$ order. There are three access paths:

   A1. Access the sorted file for $Customer$ directly.

   A2. Use a clustered B+ tree index on attribute $Customer.ID$.

   A3. Use a clustered hash index on attribute $Customer.ID$.

   For each of the following selection queries, state which of the three access paths is most likely to be the cheapest and explain why.

   (a) $\sigma_{ID=500}(Customer)$ **(1 pts)**

   (b) $\sigma_{ID\neq 500}(Customer)$ **(1 pts)**

   (c) $\sigma_{ID>500 \wedge ID<1500}(Customer)$ **(1 pts)**

2. Consider the following relational schema and SQL query. The schema captures about easter eggs in nests.

   Nest (nid,child,material)
   EggInNest (nid,eid,position)
   Egg (eid,color,size)

   EggInNest[nid] $\subseteq$ Nest[nid]
   EggInNest[eid] $\subseteq$ Egg[eid]

   Assume that each Egg record is 10 bytes long, each EggInNest record is 50 bytes long, and each Nest record is 200 bytes long. There are 5,000 tuples in Egg, 100,000 tuples in EggInNest, and 200 tuples in Nest. Each nest, identfied by nid, contains 5 eggs on average. The file system supports 2000 byte pages, and 7 buffer pages are available. All following questions are based on this information. You can assume uniform distribution of values. State any additional assumptions which you do to answer the questions. The cost metric to use is the number of page I/Os. The costs for an index access is 2 in all cases. Ignore the cost of writing out the final result.

(a) Compute the number of pages for each relation. **(3 pts)**

(b) Consider the following query:

```
SELECT *
FROM Egg E, EggInNest I
WHERE E.eid = I.eid
```

i. Compute the costs for the query using a block-nested loop join. **(2 pts)**

ii. Suppose that there is a clustered hash index on eid on Egg. Compute the costs for the query using an index-nested loop join. **(2 pts)**

iii. Assume that both relations are sorted on the join column. Which join method should be applied and what are the costs? **(2 pts)**

Exercise 5.2 (I/O Costs of Access Plans) (6 pts)

Referring to Slide 31 of Chapter 2, please reason below formulas **in detail**:

1. Why is the cost for a Range Selection using a clustered tree index $D * (\log_G 0.15B + \sharp matchingpages)$?
2. Why is the cost for an equality selection using an unclustered hash index $2D$?
3. Why is the cost for a delete operation using an unclustered tree index $D * (3 + \log_G 0.15B)$?

Base your explanation on the below assumptions from empirical studies:

- In a sorted file, pages are stored sequentially, retrieving a desired page directly only needs one disk I/O.
- In a clustered file, pages are usually 67% full, and the number of physical data pages is 1.5B.
- We omit the time for processing a record in memory (since it is usually negligible compared with the time for reading or writing disk pages)

Exercise 5.3 (Query Optimization) (12 pts)

Given a relational table Staff(sno,name,salary,marstat,dno) which is stored in an unsorted heap file with 1,000 pages (primary key is sno). Your system should be optimized for the following queries:

1. `Q1: SELECT * FROM Staff WHERE sno = 1021`
2. `Q2: SELECT name,salary FROM Staff WHERE salary > 40000 AND salary < 50000`
3. `Q3: SELECT dno, AVG(salary) FROM Staff WHERE marstat = 'single' GROUP BY dno`

How do you physically organize your database? Which indexes(clustered/unclustered) should be created to optimize the overall performance for all three queries? What are the estimated costs for your solution based on the information on Slide 31 of Chapter 2 (for the fan-out of tree index G we take 100)?