Assignment 4

Deadline: 04.06.2021

The questions that includes copy paste from internet (or anywhere) will get 0 points, even if answer is right.

1- What are the advantages and disadvantages of B+ tree for using it for indexing in databases?

2- What are the advantages and disadvantages of hashing for using it for indexing in databases?

3- Indices speed query processing, but it is usually a bad idea to create indices on every attribute, and every combination of attributes, that are potential search keys. Explain why.

4- Is it possible in general to have two clustering indices on the same relation for different search keys? Explain your answer.

5- Suppose you have a relation *r* with n_r tuples on which a secondary B$^+$-tree is to be constructed.

   a. Give a formula for the cost of building the B$^+$-tree index by inserting one record at a time. Assume each block will hold an average of *f* entries and that all levels of the tree above the leaf are in memory.

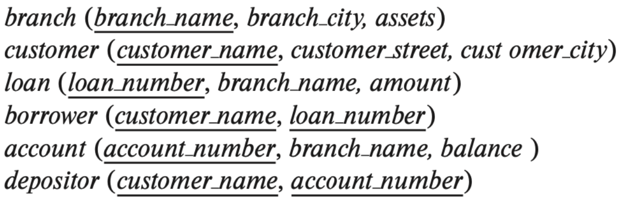
   b. Assuming a random disk access takes 10 milliseconds, what is the cost of index construction on a relation with 10 million records?

6- Suppose there is a relation *r*(*A*,*B*,*C*), with a B$^+$-tree index with search key (*A*, *B*).

   a. What is the worst-case cost of finding records satisfying $10 < A < 50$ using this index, in terms of the number of records retrieved n_1 and the height *h* of the tree?

   b. What is the worst-case cost of finding records satisfying $10 < A < 50 \wedge 5 < B < 10$ using this index, in terms of the number of records n_2 that satisfy this selection, as well as n_1 and *h* defined above?

7- Given the relations listed below,



branch (*branch_name*, *branch_city*, *assets*)
customer (*customer_name*, *customer_street*, *cust omer_city*)
loan (*loan_number*, *branch_name*, *amount*)
borrower (*customer_name*, *loan_number*)
account (*account_number*, *branch_name*, *balance*)
depositor (*customer_name*, *account_number*)

Let us define a view *branch cust* as follows:

create view *branch cust* **as**
select *branch name, customer name*
from *depositor, account*
where *depositor.account number = account.account number*

a- Suppose that the view is *materialized*; that is, the view is computed and stored. Write triggers to *maintain* the view, that is, to keep it up-to-date on insertions to *depositor* or *account*. It is not necessary to handle deletions or updates. Note that, for simplicity, we have not required the elimination of duplicates.

b- Write an SQL trigger to carry out the following action: On **delete** of an account, for each customer-owner of the account, check if the owner has any remaining accounts, and if she does not, delete her from the *depositor* relation.

8- Give characteristics of NoSQL. What is the difference between SQL and NoSQL?