Exercises on relational operators Part 2

Data Management

A.Y. 2018/19

Maurizio Lenzerini

Exercise 1

Consider the relation CONCERT(band,year,city,cost,people) that stores in a heap file information about concerts, with the band that gave the concert, the year when the concert was held, the city where the concert was held, the cost of the concert, and the number of people that attended the concert. The relation occupies 300.000 pages, each of 800 KB. We assume that all fields and pointers in any record have the same size of 10 KB, independently of the field type. There is a tree-based index on CONCERT with search key ⟨band,year,city⟩, using alternative 2. Consider the query

```
select band,city
from CONCERT
```

and tell which algorithm you would use for executing the query, and how many page accesses such algorithm needs for computing the answer in the following three cases: (1) the index is a B+-tree, and sparse, (2) the index is a B+-tree, and dense, (3) the index is dense and is an ISAM.

Exercise 2

Consider the relation

WORK(employeeNo,lastName,firstName,birthYear,company,salary)

stored in a heap file with 850.000 tuples. Assume that the size of each value is 10 Bytes, the size of each page is 600 Bytes, and the number of different companies in the relation WORK is no more that 2.700. Suppose we have 300 free buffer frames available.

- Consider the query

  select max(salary) from WORK group by company

  computing, for each company, the maximum salary given to its employees. Describe in detail the algorithm you would use to compute the answer to the query and the cost of the execution of such algorithm in terms of number of page accesses.
- Consider the query

  select * from WORK order by employeeNo

  and describe in detail the algorithm you would use to compute the answer to the query and the cost of the execution of such algorithm in terms of number of page accesses.

Exercise 3

Assume we have the relations

P1(cno, lastName, firstName, age, dateBirth, cityBirth, region)

P2(cno, lastName, firstName, age, dateBirth, cityBirth, region)

storing in heap files information about customers of two Internet providers. Each relation has 50.000.000 tuples, and each page contains 100 tuples. There are 20 processors in the network, and both relations are uniformly distributed in such a way that the tuples of each of the 20 existing regions is stored in a specific node of the network.

We have to answer the following query:

```
select *
from P1
where T.region = “NorthEast” and T.age > 30
union
select *
from P2
where T.region = “NorthEast” and T.age > 30
```

Assuming that 1/3 of the tuples with region = “NorthEast” of both relations has age =30, and that half of the nodes has 300 buffer frames available, and the other half has 500 buffer frames available, tell which is the algorithm that we should use to answer the query, and tell which is the elapsed time to compute the answer.

Exercise 4

Consider the relation TRAVEL(code,person,nation,cost) that stores in a heap file information about travels of people, with the code of the travel, the person who traveled, the nation visited, and the cost of the travel. The relation has 3.000.000 tuples, stored in 300.000 pages, and has 10.000 different values in the attribute “cost”, uniformly distributed in the relation. We assume that all fields and pointers have the same length, independently of the attribute. There is a sparse, clustering B+-tree index on TRAVEL with search key “cost”, using alternative 2. Consider the query

```
select code, nation
from TRAVEL
where cost >= 1500 and cost <= 1510
```

and tell how many page accesses we need for computing the answer to the query.

Exercise 5

Suppose that we have only 3 buffer frames available, and we have to answer the query

```
select * from R1
except
select * from R2
```

where R1 and R2 have 5000 and 9000 pages respectively, and are stored in heap files that may contain duplicates, and the result is without duplicates. Which is the most efficient algorithm for answering the query? Which is the cost of the algorithm in terms of number of page accesses?

Esercise 6

Assume that relation R(A,B) has 10.000 tuples stored in a heap file, relation Q(C,D,E,F) has 400.000 tuples, attribute D has 2.000 values uniformly distributed on the tuples of Q, each page of our system contains 400 Bytes, every attribute value or pointer requires 20 Bytes, and we have 252 free buffer frames. Consider the query

```
select A
from R
where not exists (select * from Q where Q.D = R.B)
```

and tell which is the algorithm you would use and the corresponding cost (in terms of number of page accesses) for executing such query for each of the following methods for representing Q:

(1) heap file;
(2) sorted file with sorting key D;
(3) heap file with unclustering, dense sorted index with duplicates with search key D (strongly dense index);
(4) sorted file with clustering, dense sorted index without duplicates with search key D;
(5) sorted file with clustering, sparse sorted index with search key D.