Åbo Akademi University

Advance Course on Database

Weekly Report 3

Luis Araújo(2004624)

May 27, 2021

Contents

Chapter 1

Introduction

In this weekly exercise, I was the chairman, and once the work flow was working, we decided to continue the same strategy of work. The topics approached were Hashing and bitmap indices.

Chapter 2

Exercises

2.1 1) For each of the algorithms A1 – A8 that can be used to execute select-operations, design a SQL query on the university database that could be executed with this algorithm.

A1

```
SELECT *
FROM student
WHERE dept_name = 'Finance'
```

A2

```
FROM student
USE INDEX(PRIMARY)
WHERE ID = '26080';
```

A3

```
FROM student
USE INDEX(PRIMARY)
WHERE name = 'Skeen';
```

A4

```
FROM student
USE INDEX(dept_name)
WHERE name = 'Skeen';
```

A5

```
FROM student
USE INDEX(PRIMARY)
WHERE tot_cred > 2
```

A6

```
FROM student
USE INDEX(dept_name)
WHERE tot_cred > 2
```

A7

```
FROM student
USE INDEX(dept_name)
WHERE dept_name = 'Civil Eng.' AND tot_cred > 2
```

A8

```
FROM student
USE INDEX(PRIMARY, dept_name)
WHERE dept_name = 'Civil Eng.' AND tot_cred > 2
```

2.2 2) Assume that we have a relation containing 22,000 records where all records have a fixed size of 160 bytes, and that the disk block size is 4 KB (4096 bytes). There is a B+-tree index on the primary key of the relation, and we can assume that the index is small enough to be kept in main memory at all times, so no disk accesses are needed for index lookup. How many disk seeks and block transfers can we estimate to be needed for:

2.2.1 a) A linear scan of all records in the file

In a linear scan, we will have to inspect each block of the file and test all records to see whether they satisfy the selection condition or not. So, it will be needed an initial seek to find the first block of the file, and then a read for all consecutive blocks. Therefore, let's calculate the number of existing blocks:

$22000 * 160 = 3520000\ bytes\ (Size\ of\ all\ records)$
$3520000 / 4096 \approx 860\ blocks$

We can conclude that there are 860 blocks, which means that the linear scan will make 859 block transfers.

2.2.2 b) Accessing a record with a given value on the primary key

When accessing a record with a given value on the primary key, in a B+-tree index, the algorithm will make a seek and transfer operation, to fetch the block where the record is located. So, the number of seeks and transfers will be the same as the height of the B+-tree(h) plus one.

$h + 1\ where\ h = \log_2 860 \approx 10$
$10 + 1 = 11\ seeks\ and\ transfers$

Accordingly, the selection will make 11 disk seeks and 11 block transfers.

2.2.3 c) Accessing the records with a given value on a non-key attribute

To access the records with a given value on a non-key attribute, the algorithm will have to pass through the B+-tree, just as if the given value was on the primary key, but this time the search on the disk block will not be so straight forward. Therefore, the number of seeks will be the same as the height(h) of the tree plus one, but the number of block transfers will be the height of the tree plus the number of records(b) in a block disk.

$h + b$ *where* $b = 4096 / 160 \approx 26$
$10 + 26 = 36$ *transfers*

So, there will be made 11 disk seeks and 36 block transfers.

2.2.4 d) Accessing all records with a value greater than a given value on the primary key

The difference between this selection and the previous one is that in this scan, the algorithm, after making his way through the B+-tree, it will not pass by the whole block disk, the selection will start from the point where the given value equals to the primary key. Because of this, the value can variate, depending on the given and primary key value, and so we will consider half of the size of the disk block as the number of the block transfers.

Finally, the selection will make 11 disk seeks and $10 + 13 = 23$ block transfers.

2.3 3) The relation r(A, B, C) contains 20000 tuples and the relation s(C, D, E, F) contains 45000 tuples. The tuples in r are of size 128 Bytes and the tuples in s are of size 100 bytes. The block size is 4 KB (4096 Bytes). The primary keys of the relations are underlined, and the relations have a B+-tree primary index on these keys. We assume that index blocks of the B+-tree are kept in memory and do not need to be read from disk. Explain how the following relational algebra expressions can be executed, and which of the algorithms A1 – A10 can be used. Give estimates for the number of disk seeks and transfers the queries need.

2.3.1 a) $\sigma_{C=23194}\ (r)$

Selecting c from the outer relation(r) where c = 23194. But r contains 20000 tuples as against 23194 tuples.
For this, we use algorithm A3. We retrieve all records of the relation using the index, and check each record whether the condition on a non-key attribute golds or not.
Since we have an equality on a non-key, there may be many records satisfying the criteria, but just as we assumed that the blocks of the index tree is held in memory, the cost become:

Number of seeks = 1
Number of transfers = 128

2.3.2 b) $\sigma_{C=23194}$ (s)

Selecting c from the inner relation(s) where c = 23194. And s contains 4500 tuples.
For this, we use algorithm A2. The index is used to retrieve a single record that satisfies the corresponding equality condition. Sing the index is kept in memory the cost is only one seek and one block transfer.

2.3.3 c) $\sigma_{E='Turku'}$ (s) assuming that there is a secondary B+-tree index on E in s

Selecting E from inner relation(s) where E = 'Turku'. For this query, we use algorithm A1.
The selection is equality on a candidate key attribute, once the record is found, the file scan is stopped immediately. Only one record can satisfy the selection criteria. In this kind of selection operation, linear search can be applied regardless of selection condition. This type of search algorithm (called the index scans) must be on a search-key of an index.
Number of seeks = 1
Number of transfers = 100 / 2 = 50

2.3.4 d) $\sigma_{B\geq 1992}$ (r)

Selecting B from outer relation(r) where B is greater than or equal to 1992.
For this query, we use A5 algorithm. We use the index to find the first tuple where B = 1992 and scan the relation sequentially from there. For $\sigma_{B\geq 1992}$ (r), we just need to scan the beginning of the tupple where B = 1992 to the last tuple where B ¿ 1992. If the system has information about where the relation is stored in disk no accesses to the index is needed, else, index is used to find the first block of the relation. For this selection operation:
Number of seeks = 1
Number of transfers = 128