FIB - Disseny de Bases de Dades

Access structures

Knowledge Objectives

1. Describe the different data structures (i.e., B+, Hash, Clustered index and Clustered Structure)
2. Describe the different access paths given a structure or lack of it (i.e., table scan, one tuple, several tuples)
3. Explain the difference in the cost of a modification and that of a query
4. Decide for each structure (i.e., B+, Hash, Clustered index and Clustered Structure) and operation (i.e., table scan, select one tuple, select several tuplesand join), whether it is generally worth to use the structure to perform the operation and how it would be used
5. Explain the consequences of a Clustered structure over selections of individual tables

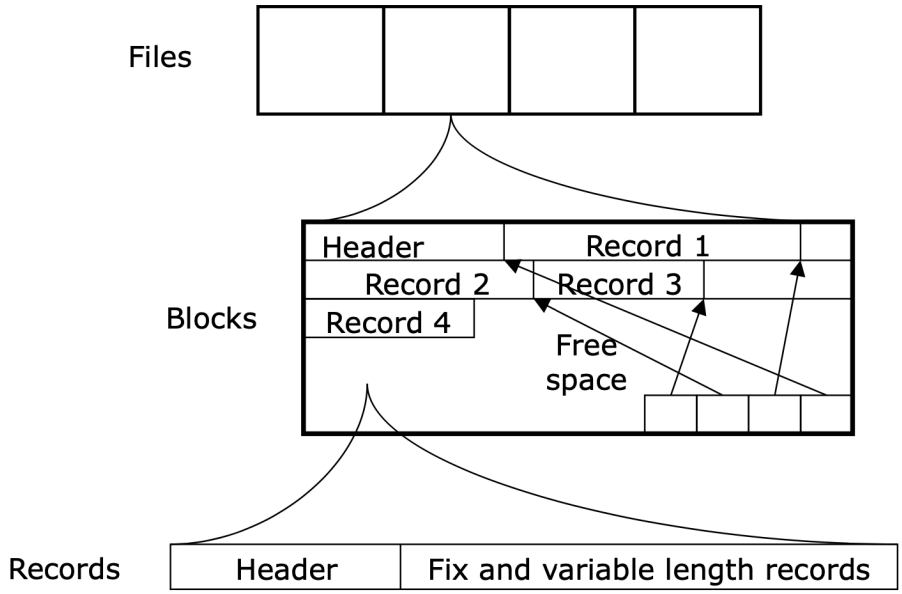
Understanding Objectives

1. Use a simplified version of cost formulas to obtain an estimation of a selection when the number of selected tuplesin known

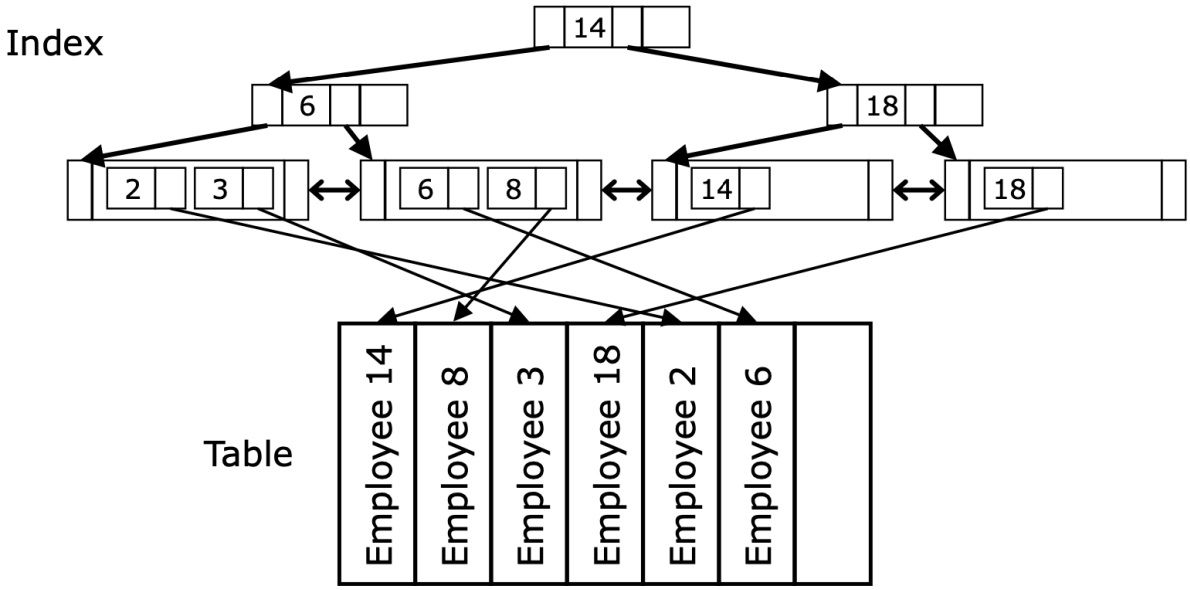
Alternatives in the structures

- Whether indexed or not, the table can be:
  - ~~Ordered~~
  - Unordered
- Two main indexing structures:
  - B-tree
  - Hash
- Indexes always keep entries composed by pairs value-information, where information can be:
  - ~~The whole record~~
  - Physical address of the record
  - ~~List of physical addresses of records~~
  - ~~Bitmap~~
- Out of all possible combinations, we will only consider:
  - No index
  - Unordered and B-tree with addresses (B+)
  - Ordered and B-tree with addresses (Clustered)
  - Unordered and hash with addresses (Hash)
  - Null values are not indexed

Table files



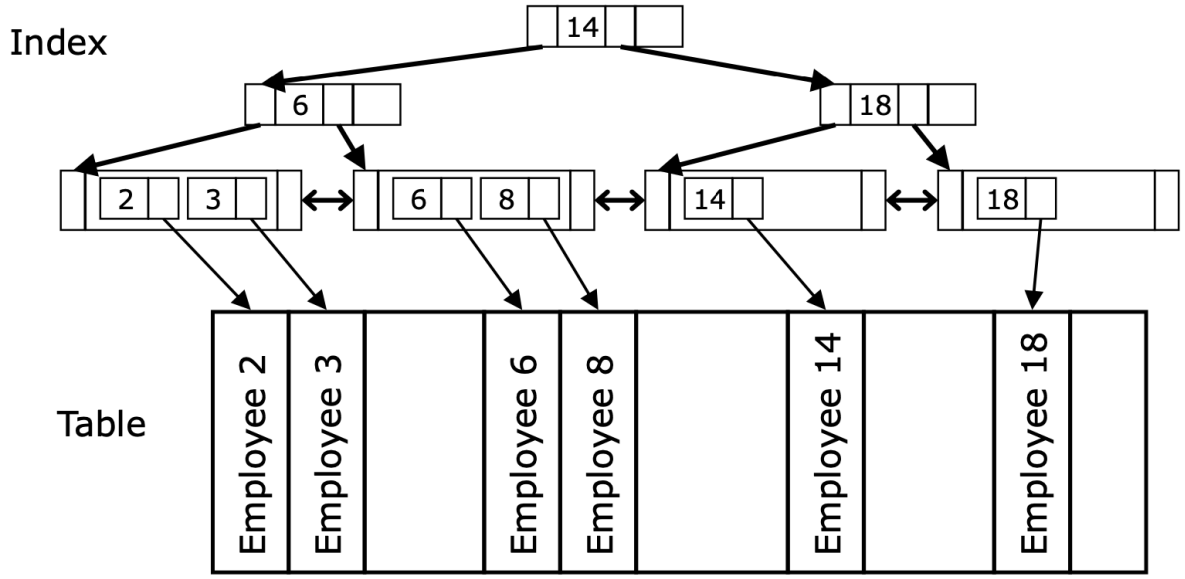
B-tree with addresses (B+)



Assumptions:

- In every tree node (a disk block) 2d addresses fit
- Usually, the tree load is 66% (2/3 of its capacity)

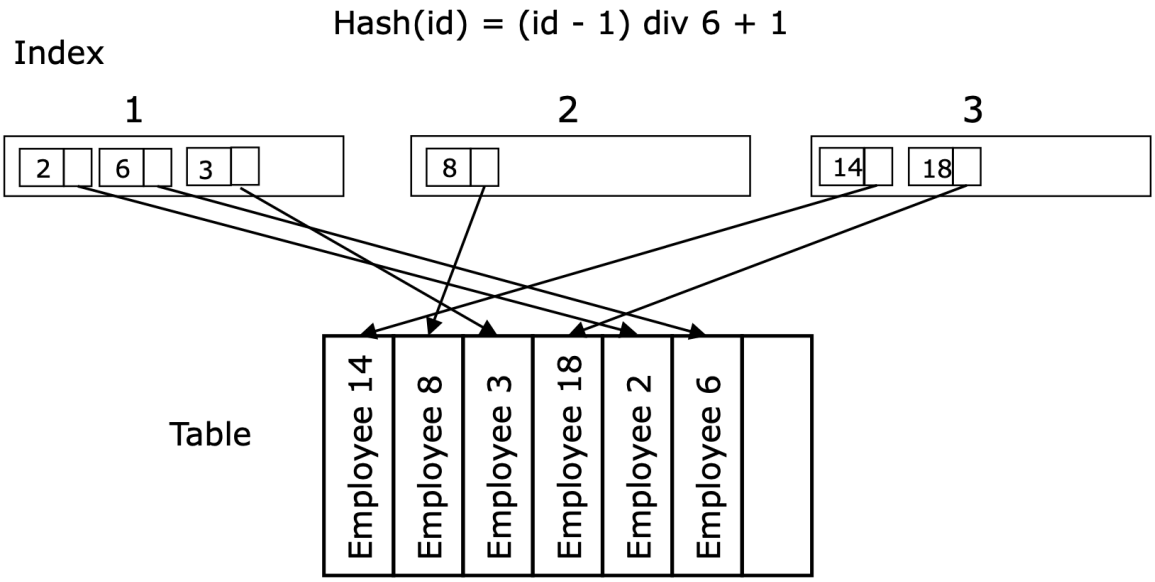
Ordered table & B-tree with addresses (Clustered)



Assumptions:

- Only 2/3 are used in every block (for index as well as for table blocks)

Hash with addresses (Hash)



Assumptions:

- There are no blocks for excess
- In a bucket block the same number of entries fit as in a tree block
- Usually, bucket blocks are used at 80% (4/5 of their capacity)

Cost variables

- B: Number of full blocks/pages that need the records
- R: Number of records per block/page
- D: Time to access (read or write) a disk block
  - Approximately 10-3 seconds (10-6 seconds for SSM)
- C: Time for the CPU to process a record

    - Approximately 10-9 seconds (we will ignore it)
- H: Time to evaluate the hash function
    - Approximately 10-9 seconds (we will ignore it)
- d: Tree order
    - Usually greater than 100
- |T|: Cardinality of table T

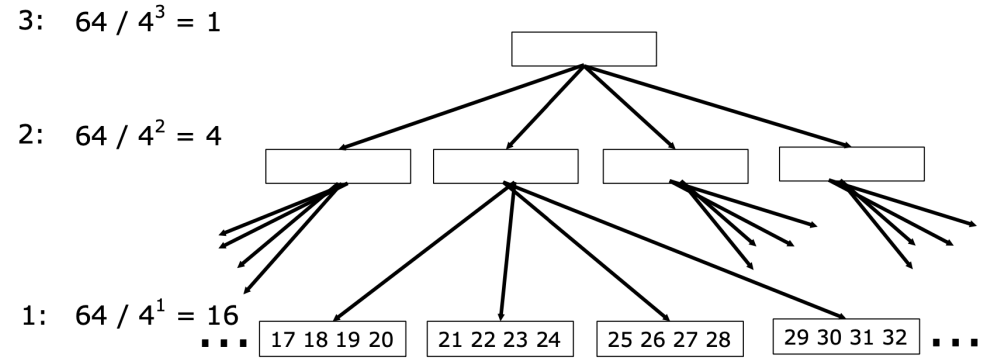
> We will not consider the improvement due to sequential scan nor cache!

Nodes and height in a B+

d = 3 load = 2/3 u = 4 (entries per node)

|T| = 64

#levels: 3 = $\log_4(64)$
h = 2
#nodes: 16 + 4 + 1 = 21



Space

- No index
    - B
- B+
    - $\sum_{1}^{h+1} \lceil |T|/u^i \rceil + B$
- Clustered
    - $\sum_{1}^{h+1} \lceil |T|/u^i \rceil + \lceil 1.5B \rceil$
- Hash
    - $1 + \lceil 1.25(|T|/2d) \rceil + B$

$u = \%load \cdot 2d = (2/3) \cdot 2d$
$h = \lceil \log_u |T| \rceil - 1$

Access paths

- Table Scan
- Search one tuple
    - Equality of unique attribute

- Search several tuples
  - Interval
  - Not unique attribute
- Insertion of one record
- Deletion of one record

Table Scan

- No index
  - B
- B+
  - $\lceil |T|/u \rceil + |T|$
- Clustered
  - $\lceil 1.5B \rceil$
- Hash
  - $\lceil 1.25(|T|/2d) \rceil + |T|$

$u = \%load \cdot 2d = (2/3) \cdot 2d$

Only useful to sort

Only useful to group

Select one tuple

- No index
  - 0.5B
- B+
  - h + 1
- Clustered
  - h + 1
- Hash
  - 2

$u = \%load \cdot 2d = (2/3) \cdot 2d$
$h = \lceil \log_u |T| \rceil - 1$

Select several tuples

- No index
  - B
- B+
  - h + (|O|-1)/u + |O|
- Clustered
  - h + 1 + 1.5(|O|-1)/R
- Hash
  - v=1: 1 + k
  - v>1: v·(1 + k)
  - v is unknown: Useless

u = %load·2d = (2/3)·2d
h = $\lceil \log_u |T| \rceil$-1
O = output table
v = values in the range
k = repetitions per value
|O|= v * k
usually computed by other means

Insertion of one tuple

- No index
  - 1 + 1
- B+
  - h + 1 + 1 + 1 (maybe more if split is needed)
- Clustered
  - h + 1 + 1 + 1 (maybe more if split or displacement are needed)
- Hash
  - 1 + 1 + 1 + 1

u = %load·2d = (2/3)·2d
h = $\lceil \log_u |T| \rceil$-1

Deletion of one record

- No index
  - 0.5B + 1
- B+
  - h + 1 + 1 + 1 (maybe more if merge is needed)
- Clustered
  - h + 1 + 1 + 1 (maybe more if merge is needed)
- Hash
  - 1 + 1 + 1 + 1

u = %load·2d = (2/3)·2d
h = $\lceil \log_u |T| \rceil$-1

Summary

- Structures
  - B+
  - Hash
  - Clustered index
- Access paths
  - Table scan
  - Select one tuple
  - Select several tuples
  - Insert one tuple
  - Delete one tuple