

Topics

- Conventional Indexes
- *B-Tree* Indexes
- *Hashing* Indexes

34

Indexes

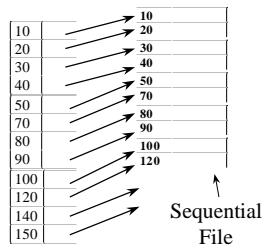
- Data structures used for quickly locating tuples that meet a specific type of condition
 - *Equality* condition:
 - *find Movie tuples where Director=\$X*
 - Other conditions possible, e.g., *range* conditions:
 - *find Employee tuples where Salary>40 AND Salary<50*
- Many types of indexes. Evaluate them on
 - Access time
 - Insertion time
 - Deletion time
 - Disk Space needed

35

Terms and Distinctions

- **Primary** index
 - the index on the attribute (a.k.a. search key) that *determines the sequencing* of the table on disk
- **Secondary** index
 - index on any other attribute
- **Dense** index
 - every value of the indexed attribute appears in the index
- **Sparse** index
 - many values do not appear

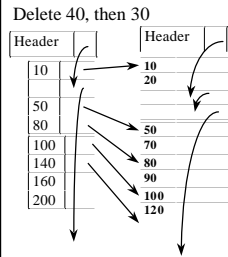
A Dense Primary Index



36

Deletion from Sparse Index (cont'd)

- if the deleted entry does not appear in the index do nothing
- if the deleted entry appears in the index replace it with the next search-key value
- ...unless the next search key value has its own index entry. In this case delete the entry



43

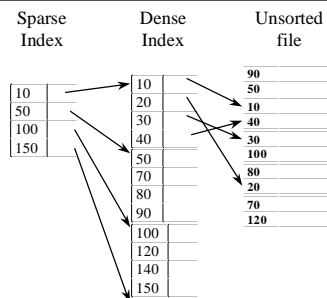
Insertion in Sparse Index

- if no new block is created then do nothing
- else create an index entry with the new value
- how to find space:
 - find nearby free space and slide blocks backward, or
 - use an overflow block

44

Secondary Indexes

- The file is not sorted according to the secondary search key
- *secondary index* has to be dense
- a *second level* index on that one would be *sparse*



45

Duplicate Values and Secondary Indexes

- store together all pointers with the same search key value

46

Duplicate Values and Secondary Indexes: Buckets

- store together all pointers with the same search key value
- introduce a separate level of buckets
 - if many pointers for each search key value it is better to separate the pointers from the values

47

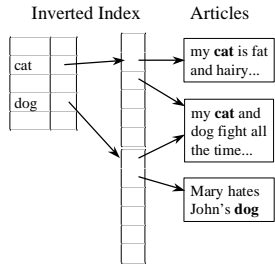
Advantage of Buckets: Process Queries Using Pointers Only

Find employees of the Toys dept with 4 years in the company
 SELECT Name FROM Employee
 WHERE Dept="Toys" AND Year=4

48

Buckets and Pointers Operation Used in Information Retrieval

- known as *inverted index*
- an entry in an inverted list represents occurrence of a word in an article
- lists range from 1 to 1,000,000 words
- compression also used



49

Summary of Indexing So Far

- Basic topics in conventional indexes
 - primary/secondary
 - sparse/dense
 - multiple levels
 - duplicate keys and buckets
 - deletion/insertion similar to sequential files
- Advantages
 - simple algorithms
 - index is sequential file
- Disadvantages
 - eventually sequentiality is lost because of overflows

50
