Database Management System – 47 (Indexing Structures for Files)

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

Outline

- Introduction
- Primary index
- Clustering index
- Secondary index

Introduction

- Indexes used to speed up record retrieval in response to certain search conditions
- Index structures provide secondary access paths
- Any field can be used to create an index
  - Multiple indexes can be constructed
- Most indexes based on ordered files
  - Tree data structures organize the index

Types of Single-Level Ordered Indexes

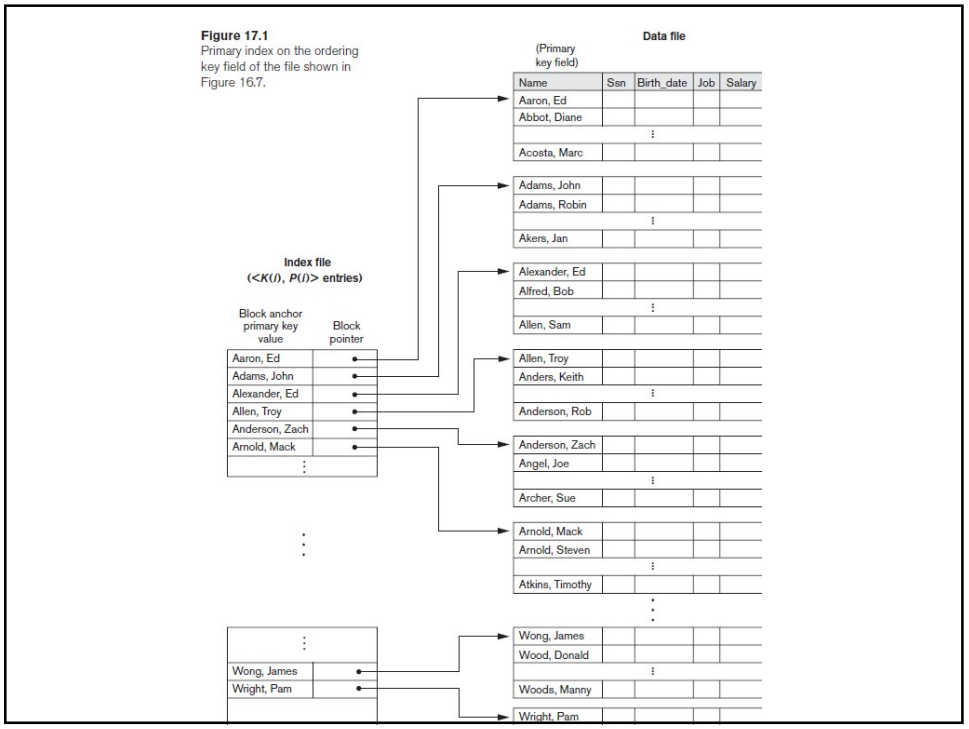
- Ordered index similar to index in a textbook
- Indexing field (attribute)
  - Index stores each value of the index field with list of pointers to all disk blocks that contain records with that field value
- Values in index are ordered
- Primary index
  - Specified on the ordering key field of ordered file of records

Types of Single-Level Ordered Indexes

- Clustering index
  - Used if numerous records can have the same value for the ordering field
- Secondary index
  - Can be specified on any nonordering field
  - Data file can have several secondary indexes

Primary Indexes

- Ordered file with two fields
  - Primary key, *K(i)*
  - Pointer to a disk block, *P(i)*
- < K (i), X >
  - X may be the physical address of a block
  - X may be the record address made up of a block address and a record id
  - X may be a logical address of the block or of the record within the file and is a relative number that would be mapped to a physical address



Primary Indexes contd...

- One index entry in the index file for each block in the data file
- First record – anchor record (block anchor)
- Indexes may be dense or sparse
  - Dense index has an index entry for every search key value in the data file
  - Sparse index has entries for only some search values

Primary Indexes contd...

- Major problem: insertion and deletion of records
  - Move records around and change index values
- Solutions
  - Use unordered overflow file
  - Use linked list of overflow records

Clustering Indexes

- Clustering field
  - File records are physically ordered on a non-key field without a distinct value for each record
- Ordered file with two fields
  - Same type as clustering field
  - Disk block pointer

  Non dense (sparse)

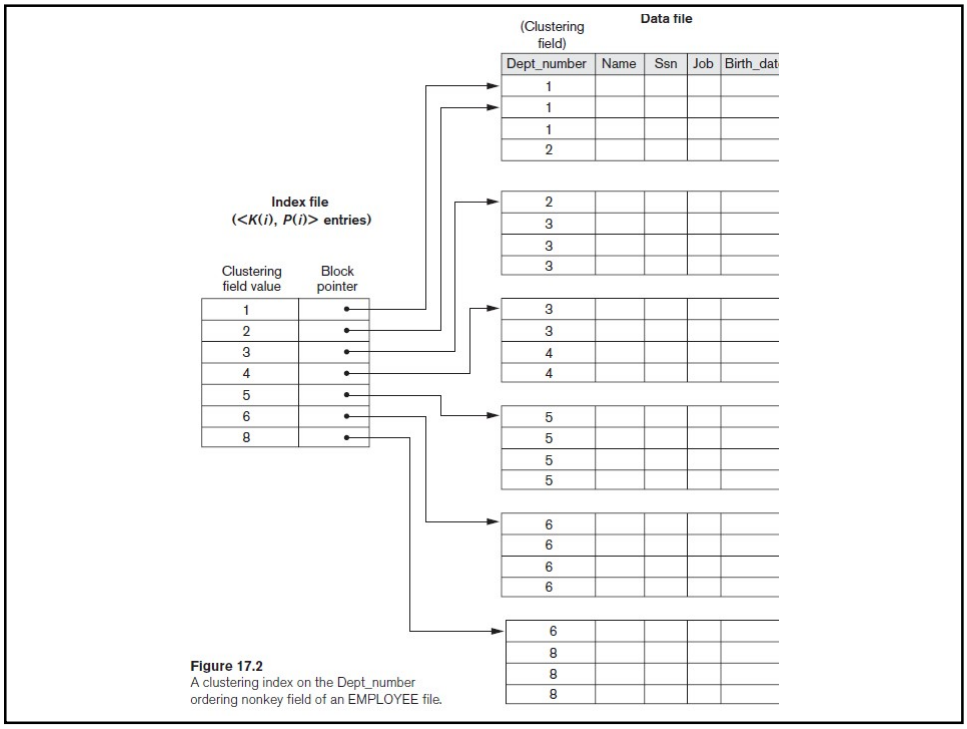


Figure 17.2
A clustering index on the Dept_number ordering nonkey field of an EMPLOYEE file.

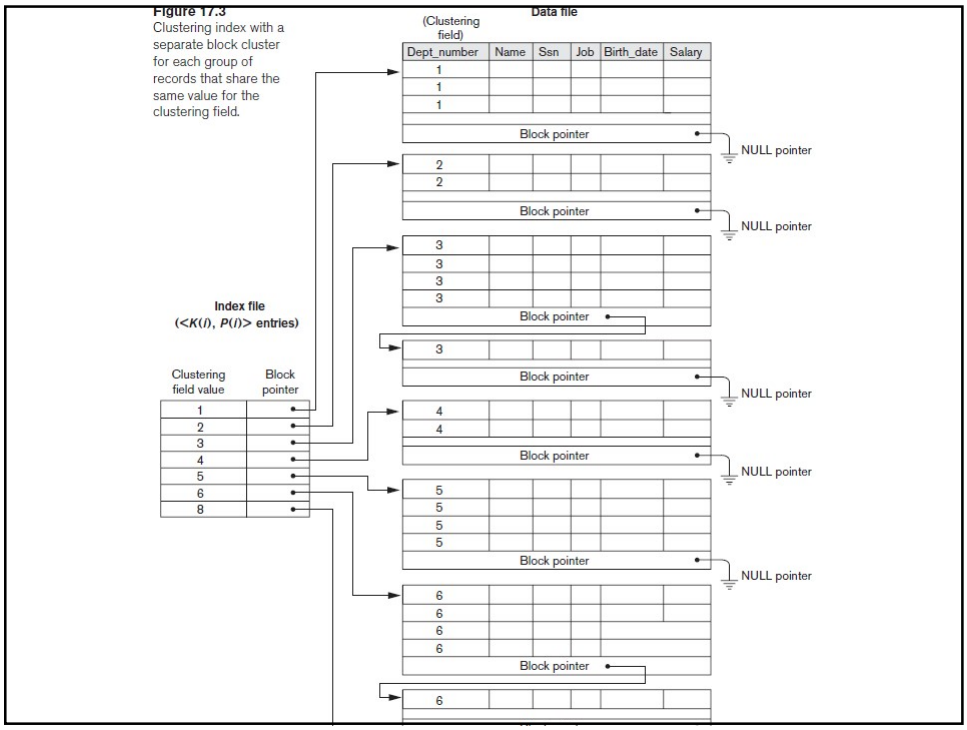


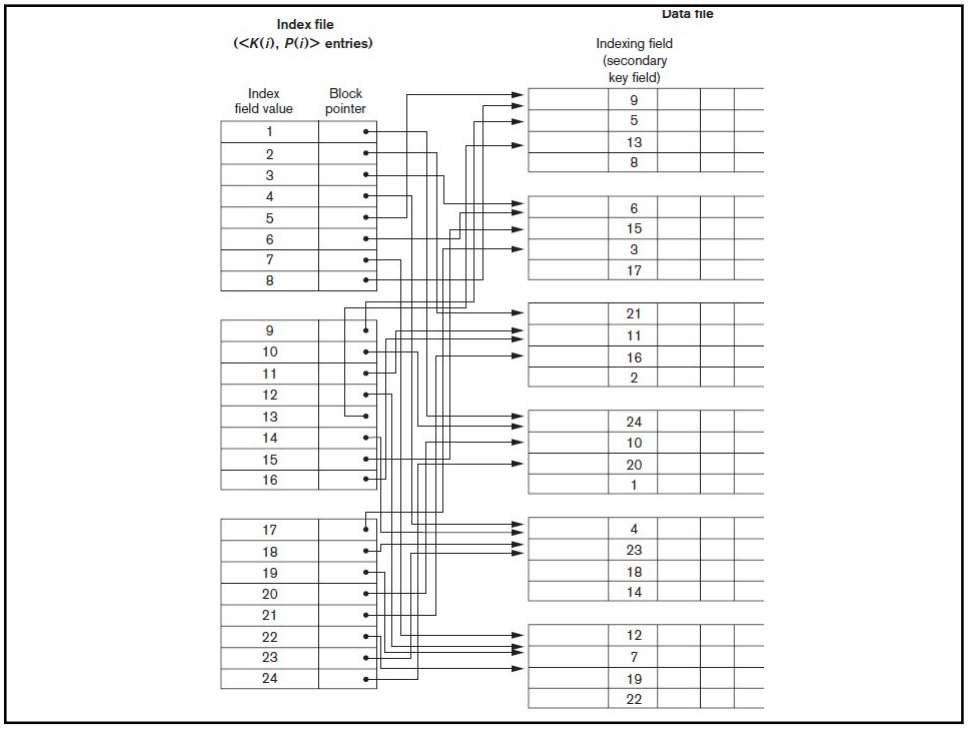
Figure 17.3
Clustering index with a separate block cluster for each group of records that share the same value for the clustering field.

Secondary Indexes

- Provide secondary means of accessing a data file
- Some primary access exists
- On candidate key (unique)
  - dense
- Or on a nonkey attribute (duplicate)
  - Sparse

Secondary Indexes

- Ordered file with two fields
  - Indexing field, *K(i)*
  - Block pointer or record pointer, *P(i)*
- Not Physically ordered (no block anchors)
- Usually need more storage space and longer search time than primary index
- Improved search time for arbitrary record

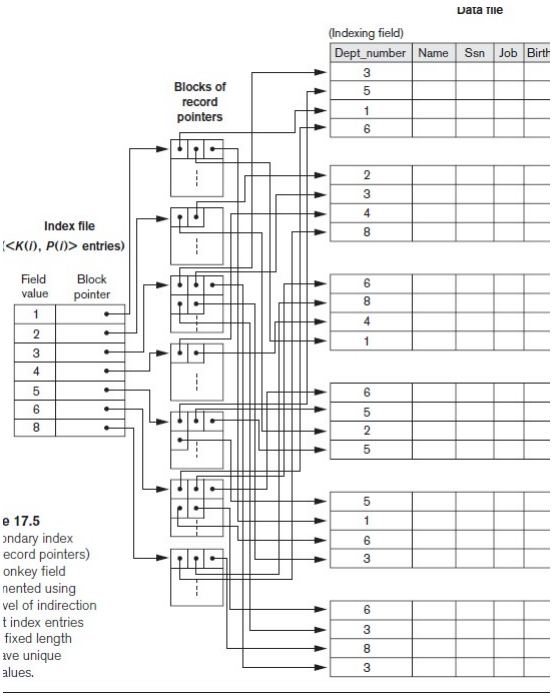


Secondary Indexes

- Can be on Nonkey, nonordering field
- Numerous records in the data file can have the same value
- Option 1
  - include duplicate index entries with the same K(i) value
  - one for each record
  - dense index.

Secondary Indexes

- Option 2
  - variable-length records for the index entries, with a repeating field for the pointer.
  - List of pointers <P(i, 1), … , P(i, k)>
- Option 3
  - keep the index entries themselves at a fixed length and have a single entry for each index field value
  - But create an extra level of indirection to handle the multiple pointers.



e 17.5
ondary index
ecord pointers)
onkey field
nented using
vel of indirection
t index entries
fixed length
ave unique
alues.

Summary

Table 17.1 Types of Indexes Based on the Properties of the Indexing Field

	Index Field Used for Physical Ordering of the File	Index Field Not Used for Phys Ordering of the File
Indexing field is key	Primary index	Secondary index (Key)
Indexing field is nonkey	Clustering index	Secondary index (NonKey)

Table 17.2 Properties of Index Types

Type of Index	Number of (First-Level) Index Entries	Dense or Nondense (Sparse)	Block Ancho on the Data
Primary	Number of blocks in data file	Nondense	Yes
Clustering	Number of distinct index field values	Nondense	Yes/no[a]
Secondary (key)	Number of records in data file	Dense	No
Secondary (nonkey)	Number of records[b] or number of distinct index field values[c]	Dense or Nondense	No

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you