Database Management Systems

BITS Pilani
Hyderabad Campus

Dr.R.Gururaj
CS&IS Dept.

Lecture Session-13

Indexing -1

innovate achieve lead

Content

- *What is Indexing*
- *Primary and Secondary indexes*
- *Dense and Sparse Indexing*
- *Multilevel Indexing*
- *Designing Primary and Multilevel Indexes*

Introduction to Indexing

An *index* for a file works in much the same way as a catalog in a library.
In a library cards are kept in alphabetical order. So we don't have to search all cards.
In real world databases, indexes may be too large to be handled efficiently.
Hence some sophisticated techniques are to be used.

Techniques for efficient retrieval of required records from disk are:

- Hashing
- Indexing

The criteria for evaluating the hashing or indexing techniques –

- Access time
- Insertion time (new indexes or new records)
- Deletion time
- space overhead

Some times more than one indexing may be required for a file.

The attribute /field used for constructing index structure for a file is called a '*indexing field/attribute*' .

If the index field is a key, it is called as search key or indexing key.

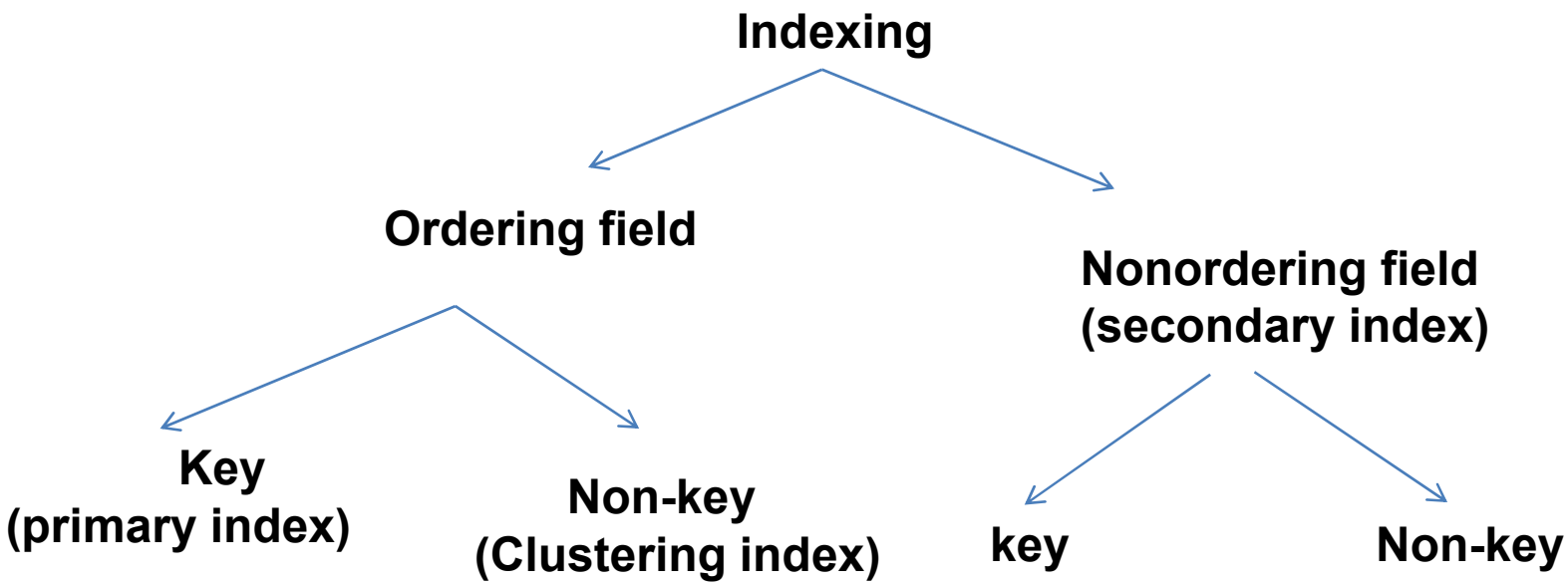
Indexes on key attributes:

1. Built on ordering key(PK) – Primary index
2. Non-ordering Key - Secondary index on key attribute

Indexes on non-key attributes:

1. Ordering non-key -- Clustering Index
2. Non-ordering non-key attribute – Secondary index on non-key

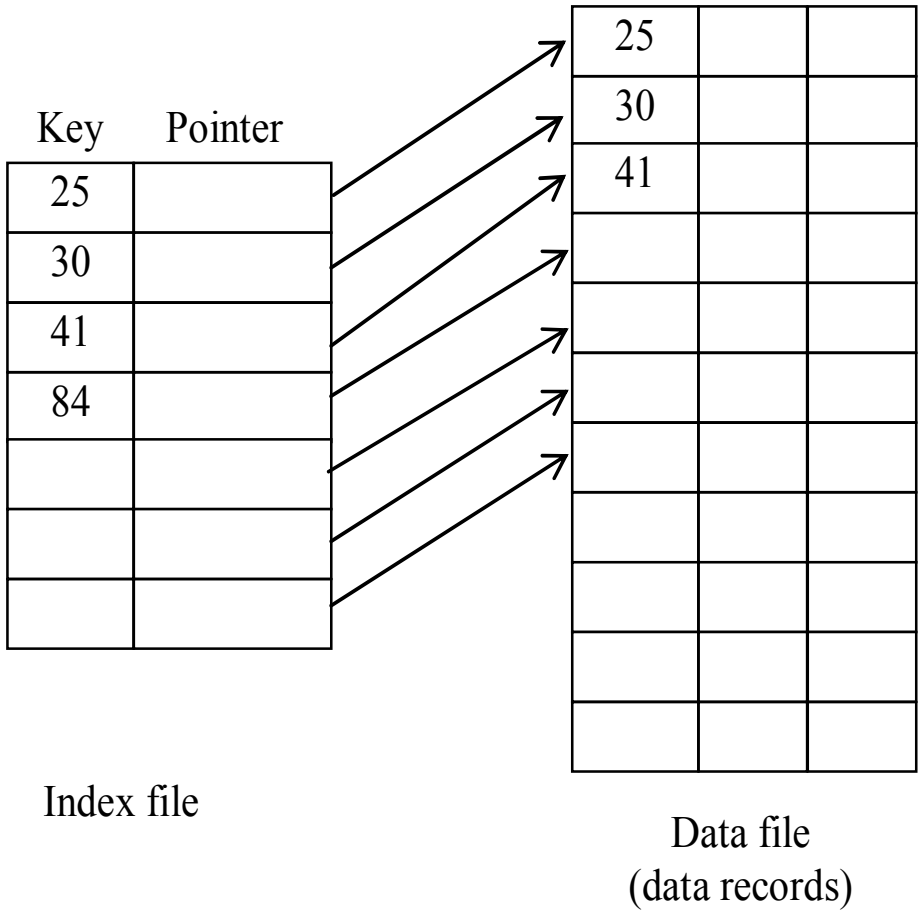
Hence, a file can have at most one primary index o one clustering index, but not both.


Indexing
Ordering field
Nonordering field
(secondary index)
Key
(primary index)
Non-key
(Clustering index)
key
Non-key

Data record: Similar kind of records(of a relation/table) are stored in a single file containing blocks. These are called data records and will have fields specified on the relation.

Index record: Like data records, index records are also stored in database. Any index record normally has two fields.

Value	Pointer
Key value	Location address of the record containing the key

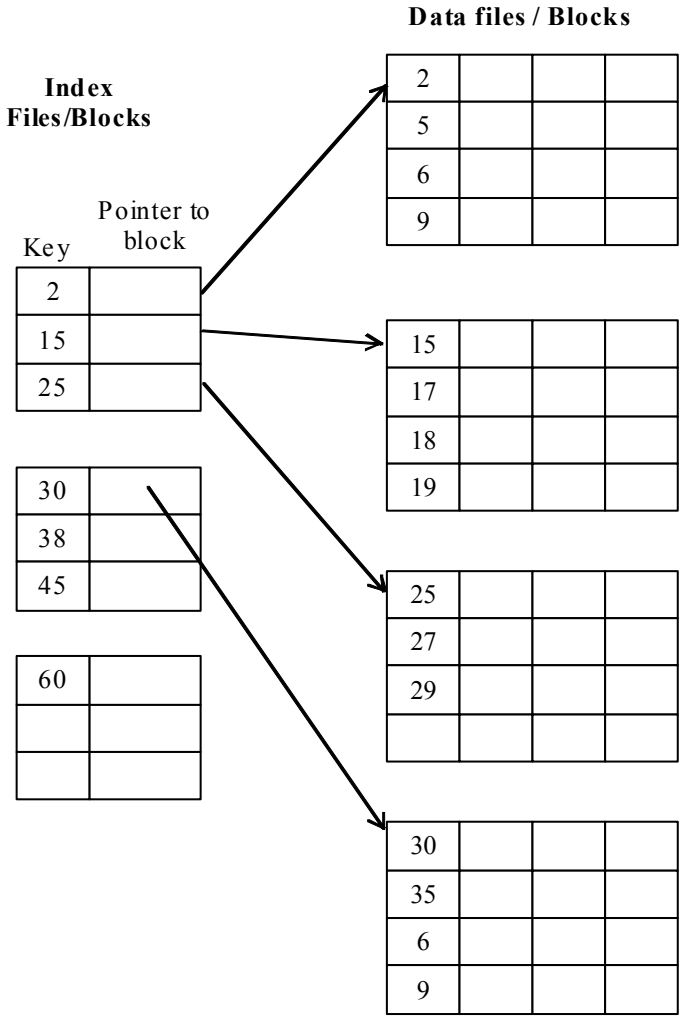

Key
Pointer
25
30
41
84
25
30
41
Index file
Data file
(data records)

Dense Index : In this, an index record appears for every data file record.

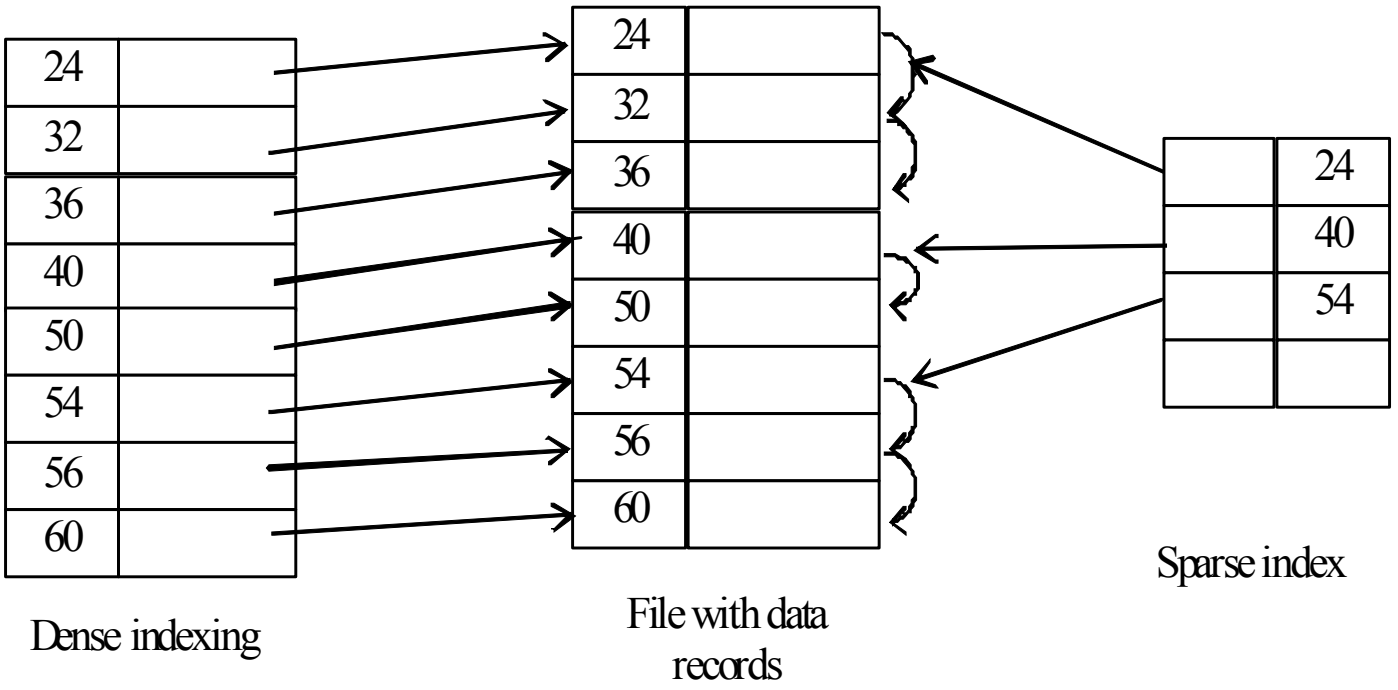
Sparse Index : Index records are created only for some data file records. This occupies less space. Sparse index can be on primary or secondary key.

A primary index and clustering index are non-dense.

Primary Indexing

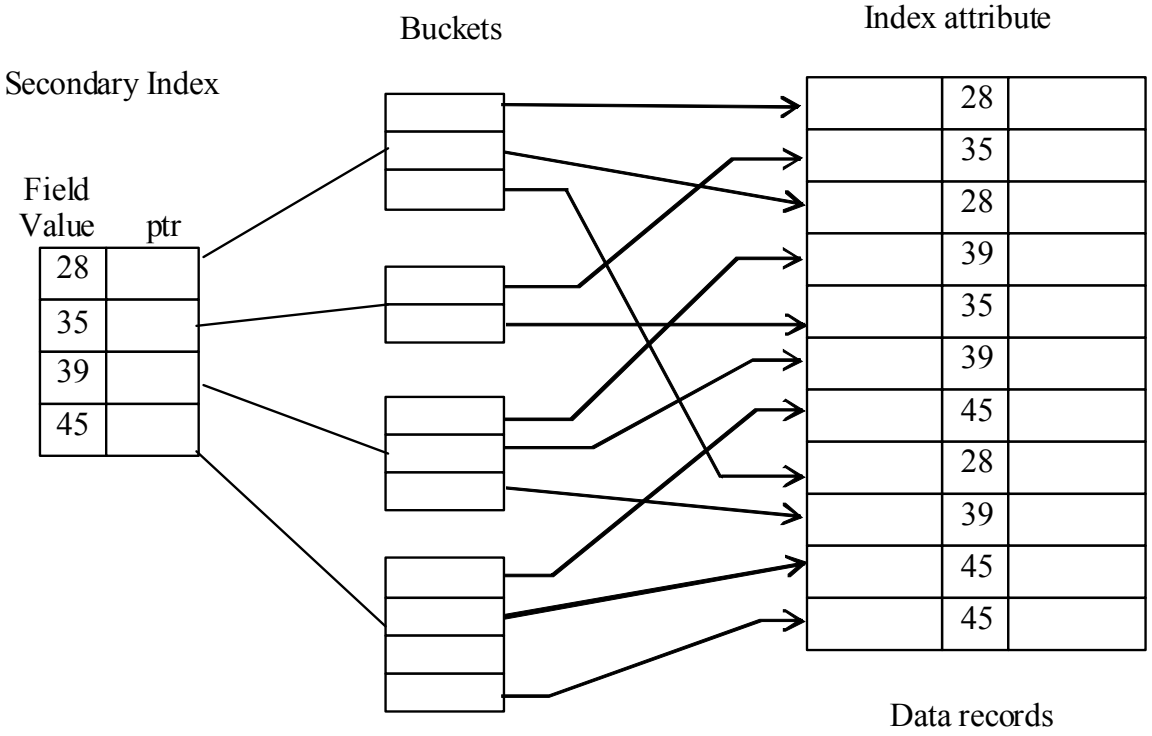


Dense and Sparse Indexing

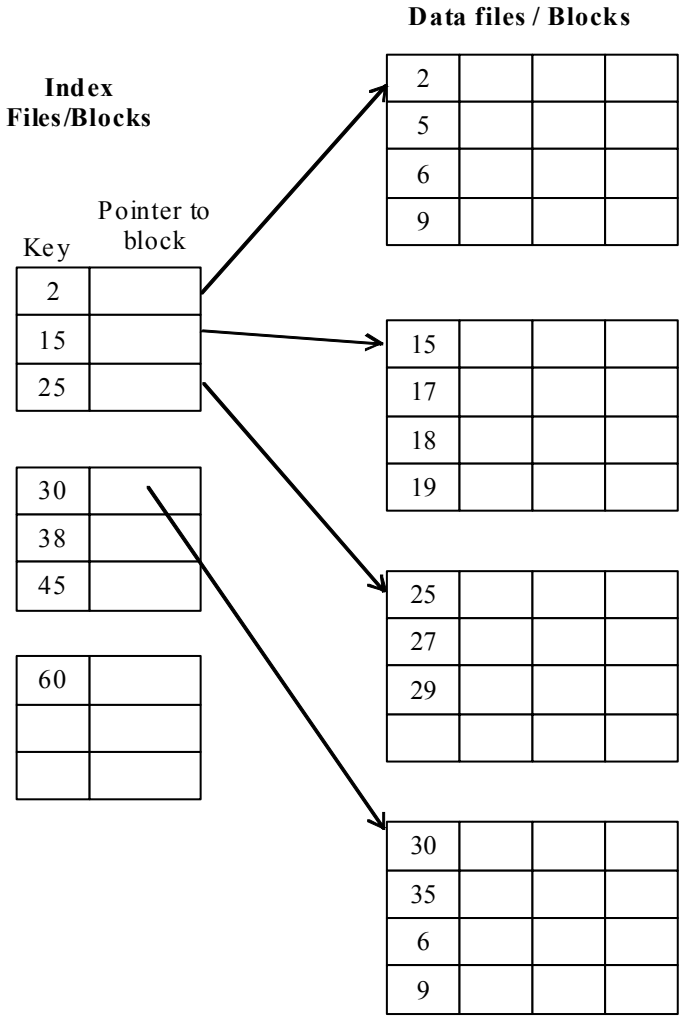


Secondary Indexing

(Built on non-ordering non-key attribute)



Primary Indexing



Designing a Primary index

Assume that we have an ordered file with 80000 records stored on disk. Block size is 512 Bytes. Record length is fixed and it is 70 Bytes. Key field(PK) length is 6 Bytes and block pointer is 4 Bytes. Assume unspanned record organization

Design a Primary index on primary key.

Sizeof disk block=512 Bytes; record length=70 Bytes

Block pointer=4 Bytes. Key field=6 bytes; total records=80000

No. records per block(*Bfr*)= floor (512/70)=7.31=7

No. of data blocks needed= ceil(80000/7)= 11429

Index record length= key + pointer=6+4=10 Bytes

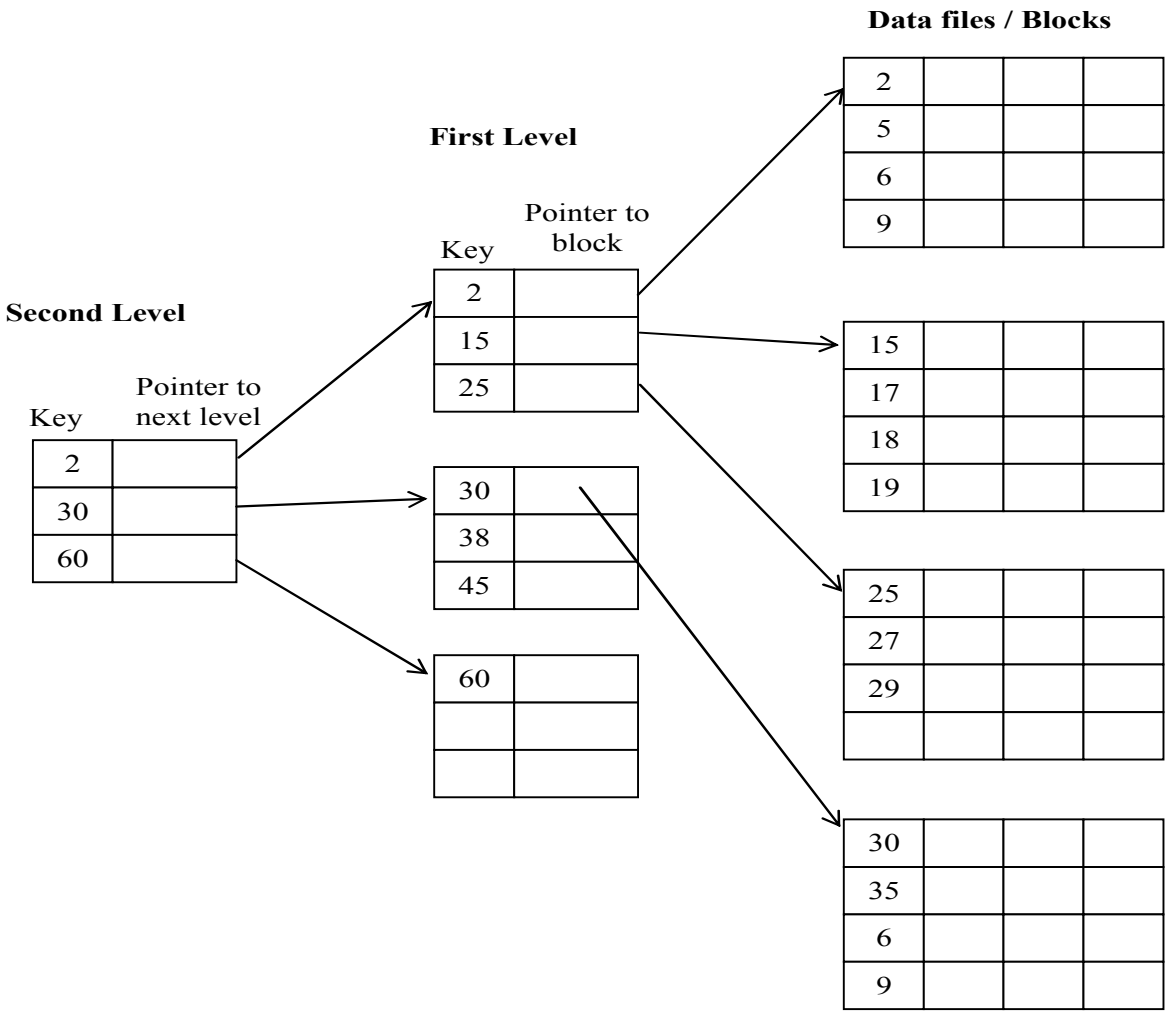
Blocking factor for index (Bfr_i) = floor(512/10)=51

(known as fanout)

No. of index blocks = Ceil(11429/51)= 225

No. of block accesses= floor of ($\log_2$ 225) + 1 = 8+1=9

Multilevel Indexing (Two levels)



Designing a multilevel index

Assume that we have an ordered file with 80000 records stored on disk. Block size is 512 Bytes. Record length is fixed and it is 70 Bytes. Key field(PK) length is 6 Bytes and block pointer is 4 Bytes. Assume unspanned record organization

Design a multilevel index on primary key.

How many levels are there.

How many blocks are there in each index level.

Size of the disk block=512 Bytes; record length=70 Bytes
Block pointer=4 Bytes. Key field=6 bytes; total records=80000
No. records per block(Bfr)= floor (512/70)=7.31=7
No. of data blocks needed= ceil(80000/7)= 11429
Index record length= key + pointer=6+4=10 Bytes
Blocking factor for index = floor(512/10)=51 - fanout
No. of index blocks in first level= Ceil(311429/51)= 225
No. of index blocks in 2nd level= Ceil(225/51)= 15
No. of index blocks in 3rd level= Ceil(5/51)= 1 top level

No. of levels=t=3
No. of block accesses= No. index levels + 1= t+1=4

Action on deletion of records

If the record is the last record with that value delete the entry in index file too. If it is dense index delete it like record in a file. If it is sparse, we delete the entry and replace with next key value, if it is not already existing.

Action on Inserting a new record

If the indexing is dense, insert the new key into the index. If sparse no change is to be made unless new block is created.

Summary

- *What is Indexing and its importance*
- *How Primary and Secondary indexes work*
- *Examples of Dense and Sparse Indexes*
- *What is Multilevel Indexing*
- *Some example problems on designing Primary and Multilevel Indexes*