Database Management System

Physical Database Design

Dr. Nachiyappan S

VIT-Chennai

CSE2004 Syllabus

Module-6

- Indexing :
  - Single Level Indexing
  - Multi-Level Indexing
  - Dynamic Multi-Level Indexing

Indexing in DBMS

What is Indexing?

- **INDEXING** is a data structure technique which allows you to quickly retrieve records from a database file.
- An Index is a small table having only two columns.
- The **first** column comprises a copy of the primary or candidate key of a table. Its **second** column contains a set of pointers for holding the address of the disk block where that specific key value stored.
- An Index is a :
  - Takes a search key as input
  - Efficiently returns a collection of matching records.

Types of Indexing...!

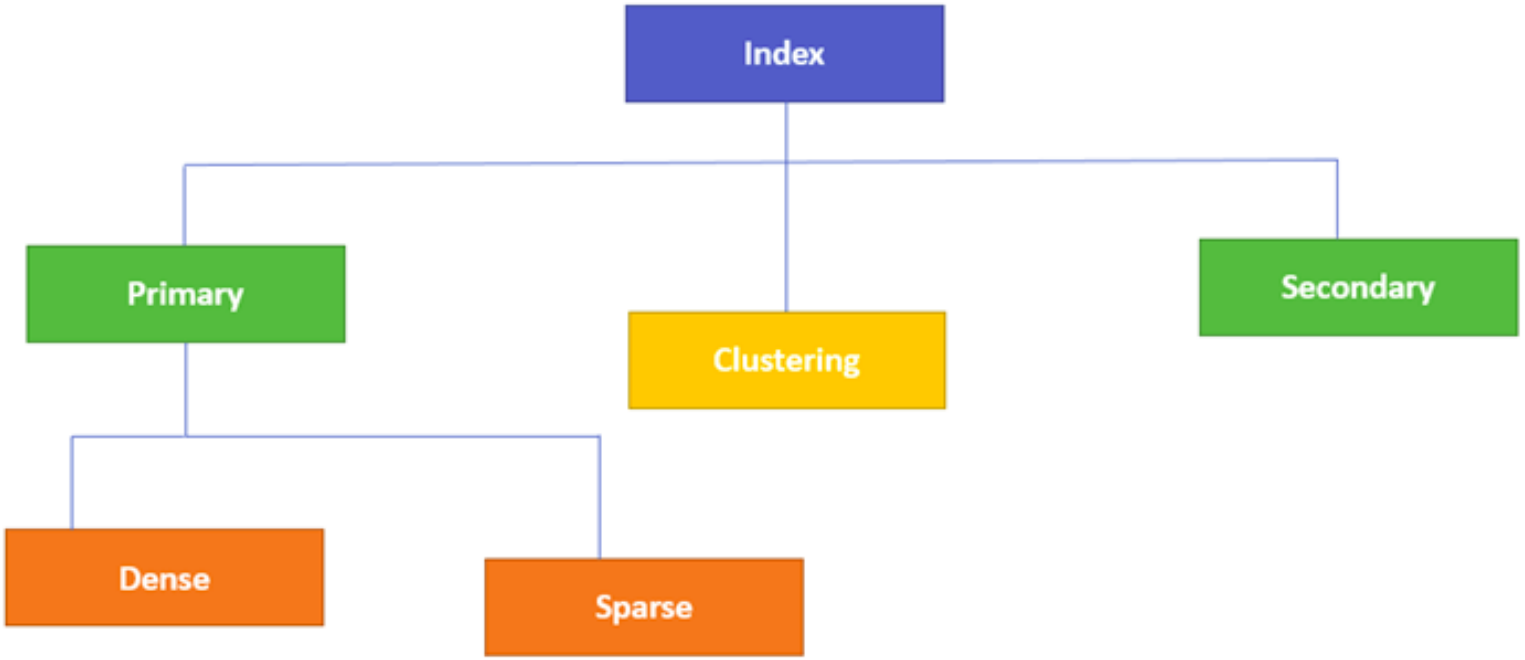


Figure: Types of Indexing

Database Indexing is defined based on its indexing attributes.
Two main types of indexing methods are:

- Primary
- Secondary

Primary Indexing

- Primary Index is an ordered file which is fixed length size with two fields.
- The first field is the same a primary key and second field is pointed to that specific data block.
- In the primary Index, there is always one to one relationship between the entries in the index table.
- The primary Indexing is also further divided into two types
  - Dense Index
  - Sparse Index

Dense Index

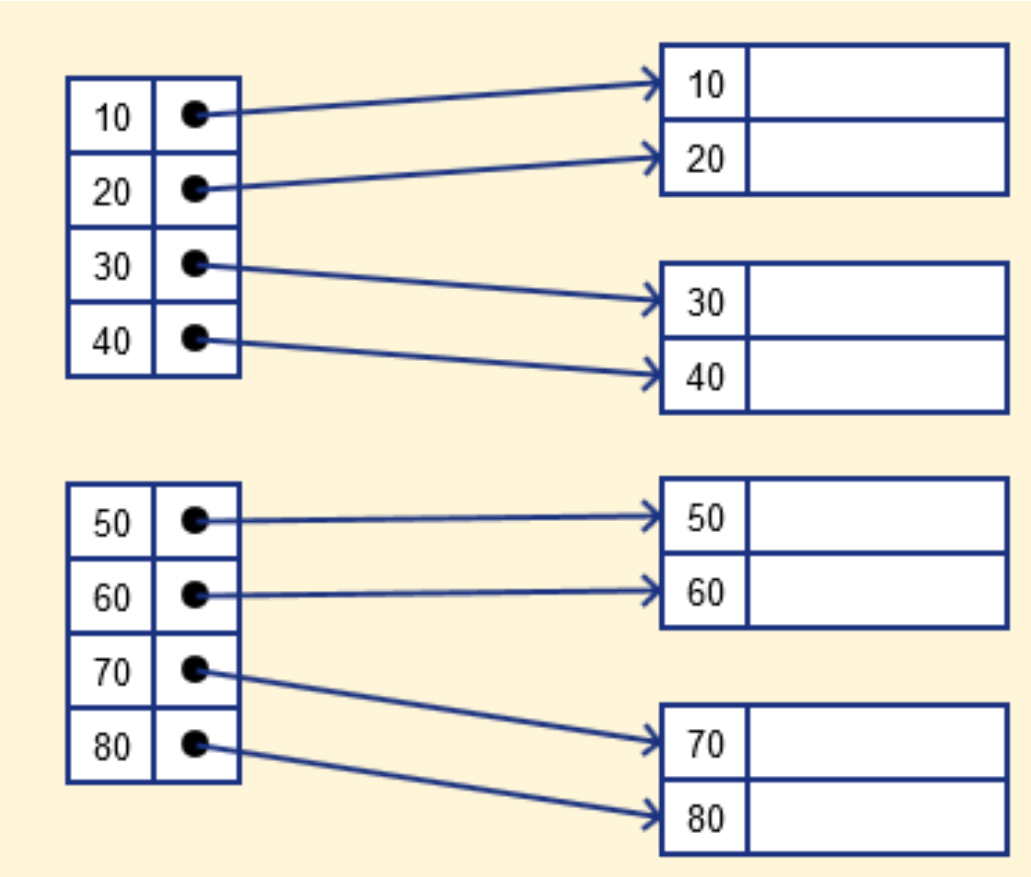
Primary Indexing

- In a dense index, a record is created for every search key valued in the database.
- This helps, to search faster but needs more space to store index records.
- In this Indexing, method records contain search key value and points to the real record on the disk.
- Example-1 :

Dense Index

Primary Indexing

Example-2 :



Sparse Index

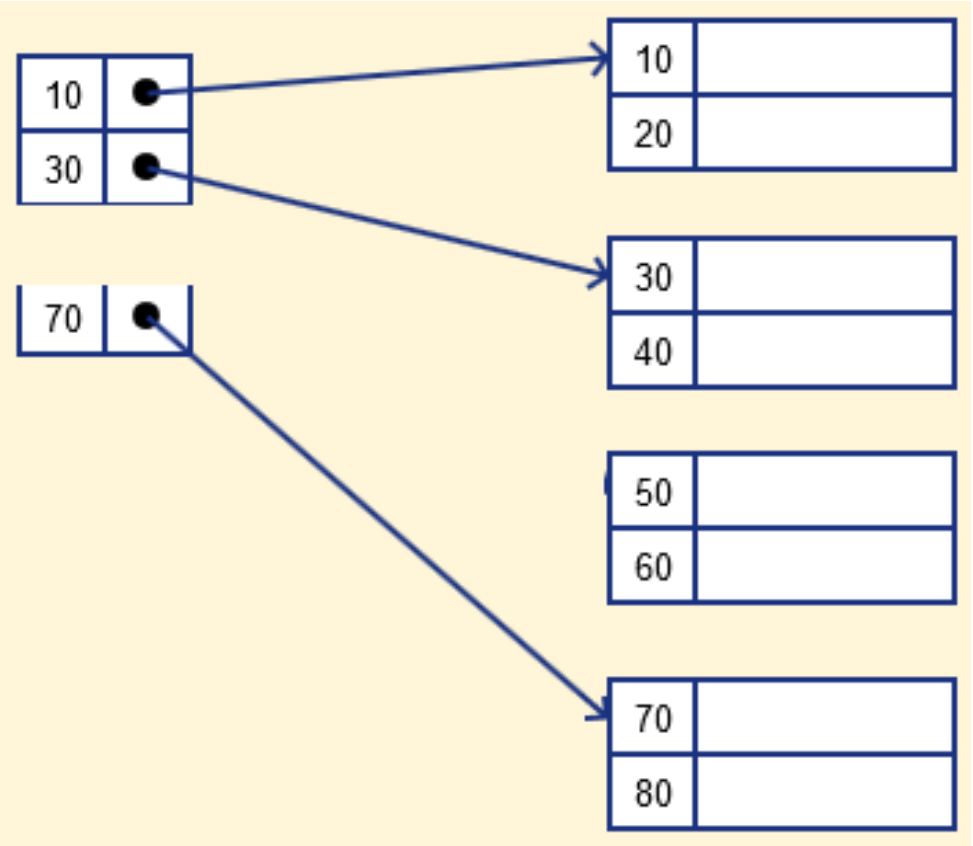
Primary Indexing

- It is an index record that appears for only some of the values in the file.
- Sparse Index helps you to resolve the issues of dense Indexing.
- In this indexing, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.
- However, sparse Index stores index records for only some search-key values.
- It needs less space, less maintenance overhead for insertion, and deletions but It is slower compared to the dense Index for locating records.
- Example-1 :

Sparse Index

Primary Indexing

- Example-2 :

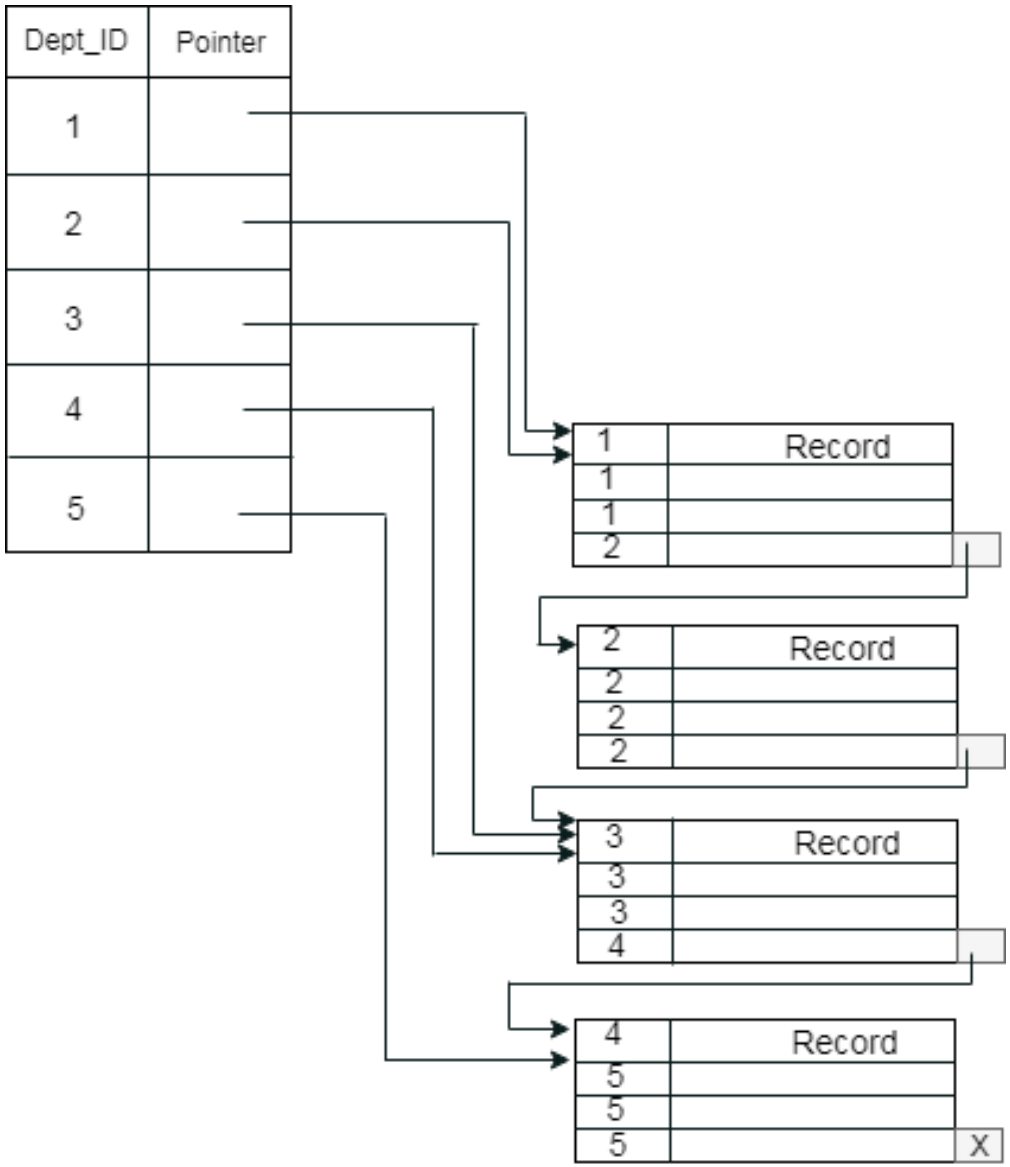


Clustering Index (Eg) Primary Indexing

- It can be defined as an ordered data file. Sometimes the index is created on non-primary key columns which may not be unique for each record.
- In this case, to identify the record faster, we will group two or more columns to get the unique value and create index out of them. This method is called a clustering index.
- The records which have similar characteristics are grouped, and indexes are created for these group.
- **Example-1** : Suppose a company contains several employees in each department. Suppose we use a clustering index, where all employees which belong to the same Dept_ID are considered within a single cluster, and index pointers point to the cluster as a whole. Here Dept_Id is a non-unique key.

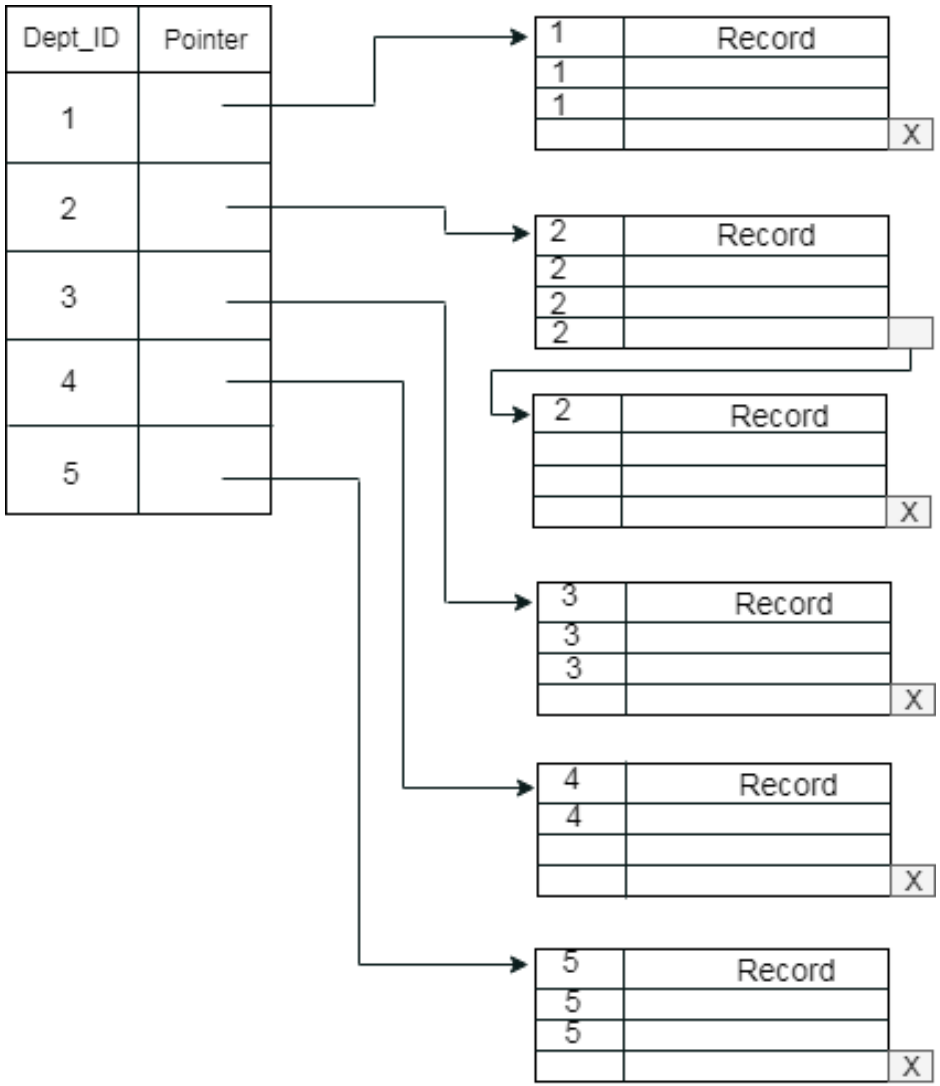
Clustering Index

- Example-1 :



Clustering Index

In Previous Schema, one disk block is shared by the records, belongs to the different cluster. If we use separate them, then it is called better technique.

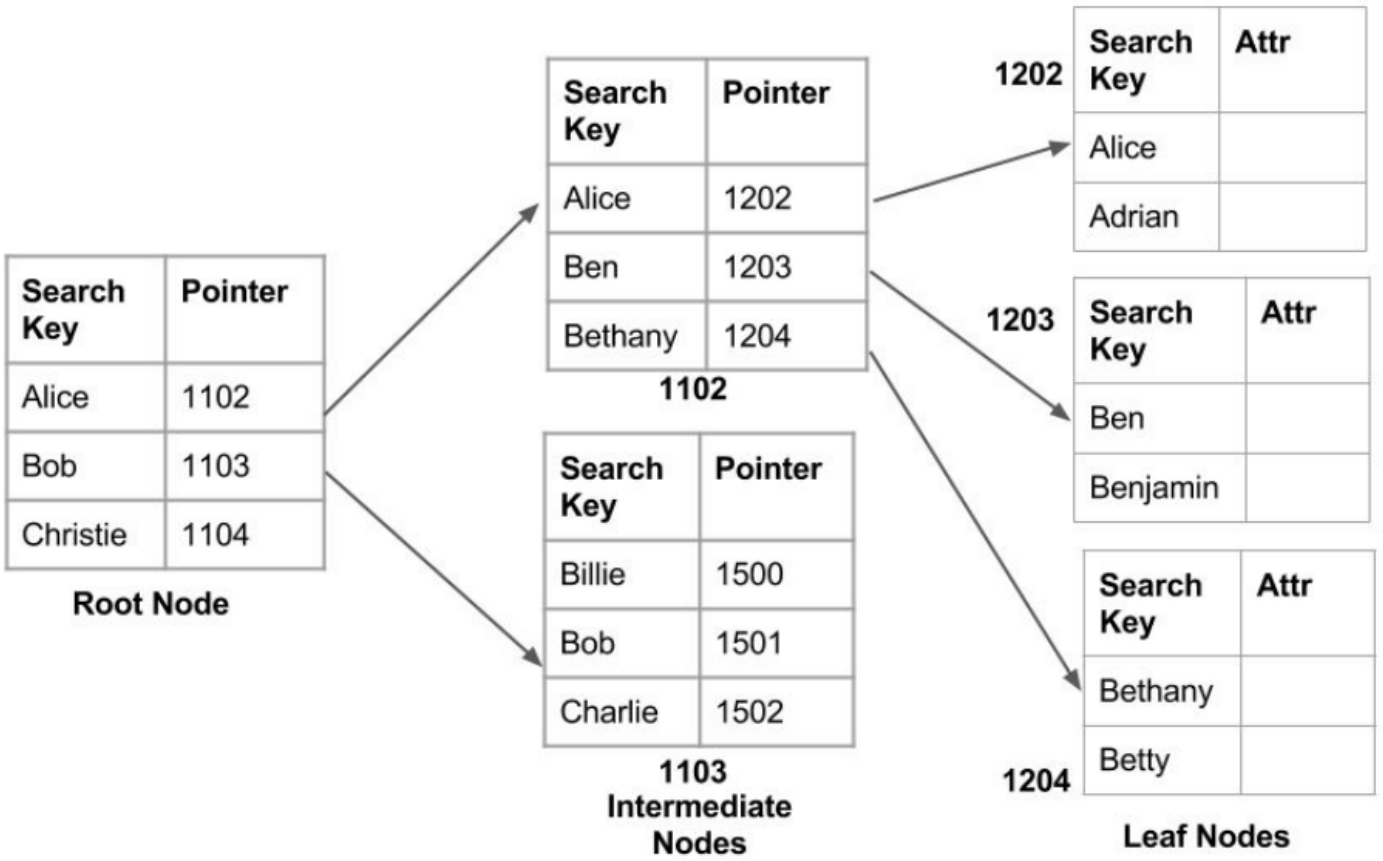


Secondary Indexing or Non-Clustering Indexing

- A non clustered index just tells us where the data lies, i.e. it gives us a list of virtual pointers or references to the location where the data is actually stored.
- Data is not physically stored in the order of the index. Instead, data is present in leaf nodes.
- For Eg. The contents page of a book. Each entry gives us the page number or location of the information stored.
- The actual data here(information on each page of the book) is not organized but we have an ordered reference(contents page) to where the data points actually lie.
- We can have only dense ordering in the non-clustered index as sparse ordering is not possible because data is not physically organized accordingly.
- It requires more time as compared to the clustered index because some amount of extra work is done in order to extract the data by further following the pointer. In the case of a clustered index, data is directly present in front of the index.

Secondary Indexing or Non-Clustering Indexing

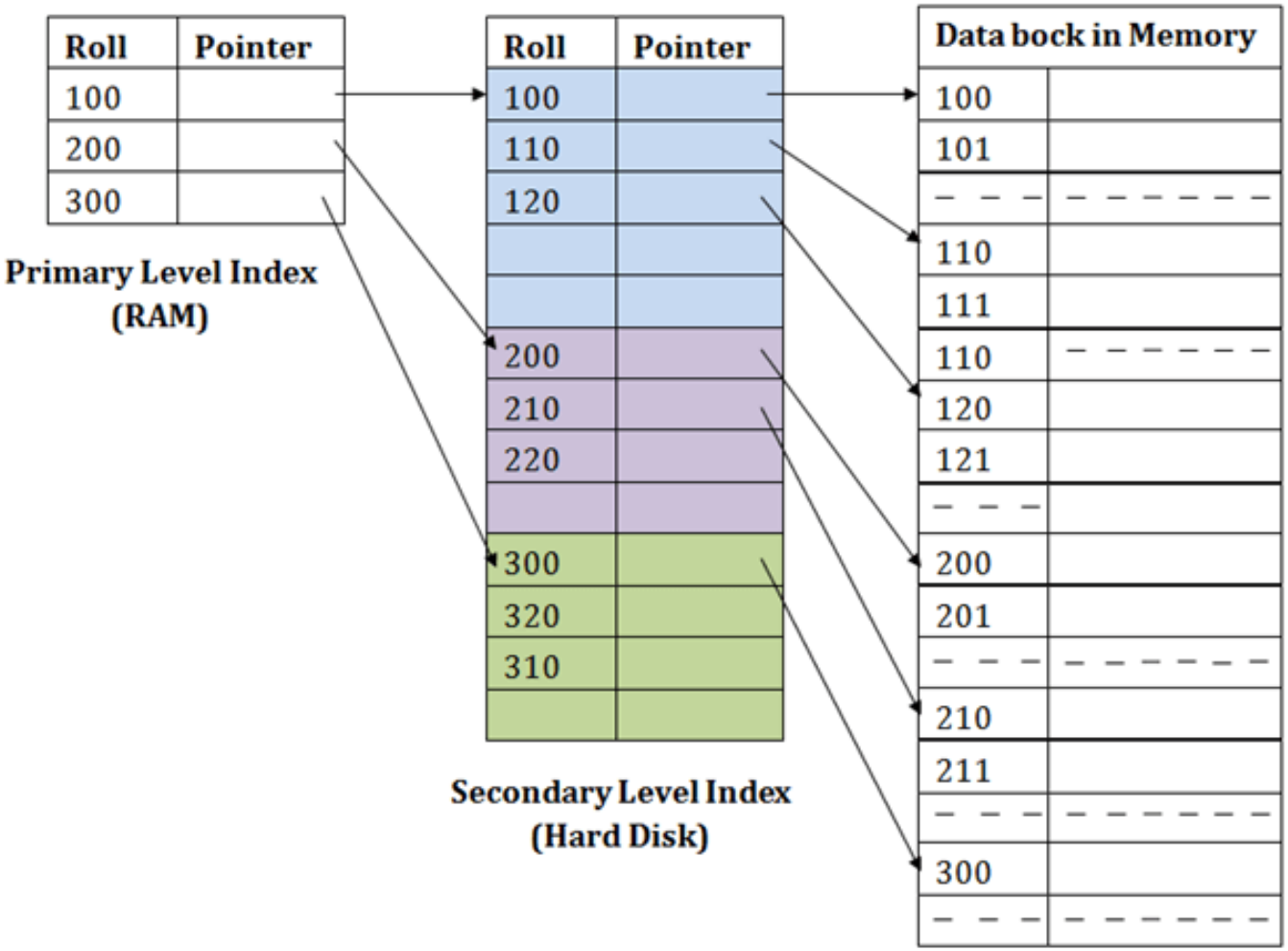
- Exampe-1:



Non clustered index

Secondary Indexing or Non-Clustering Indexing

- Exampe-2:

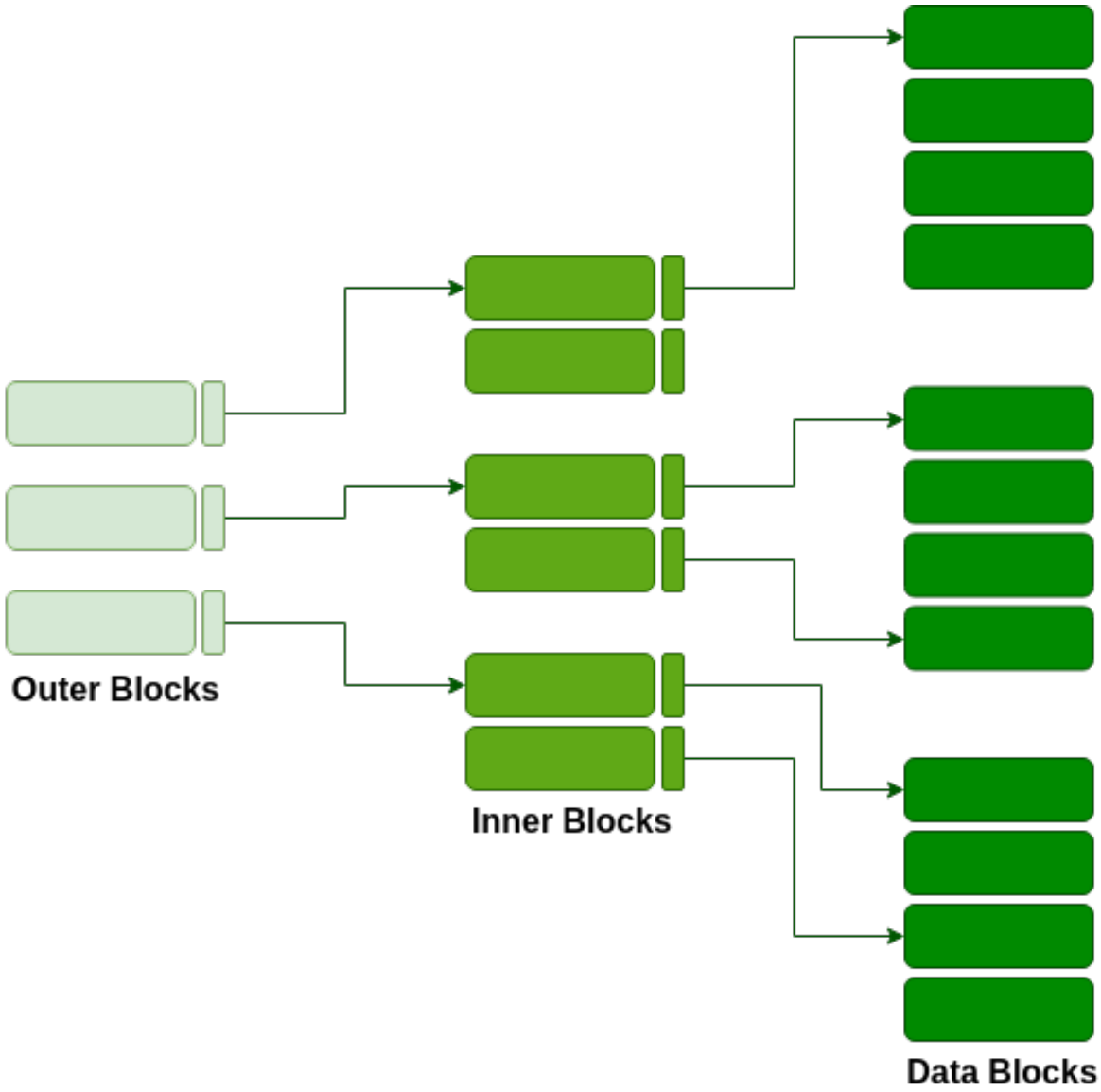


Multi-Level Indexing

- With the growth of the size of the database, indices also grow. As the index is stored in the main memory, a single-level index might become too large a size to store with multiple disk accesses.
- The multilevel indexing segregates the main block into various smaller blocks so that the same can stored in a single block.
- The outer blocks are divided into inner blocks which in turn are pointed to the data blocks. This can be easily stored in the main memory with fewer overheads.

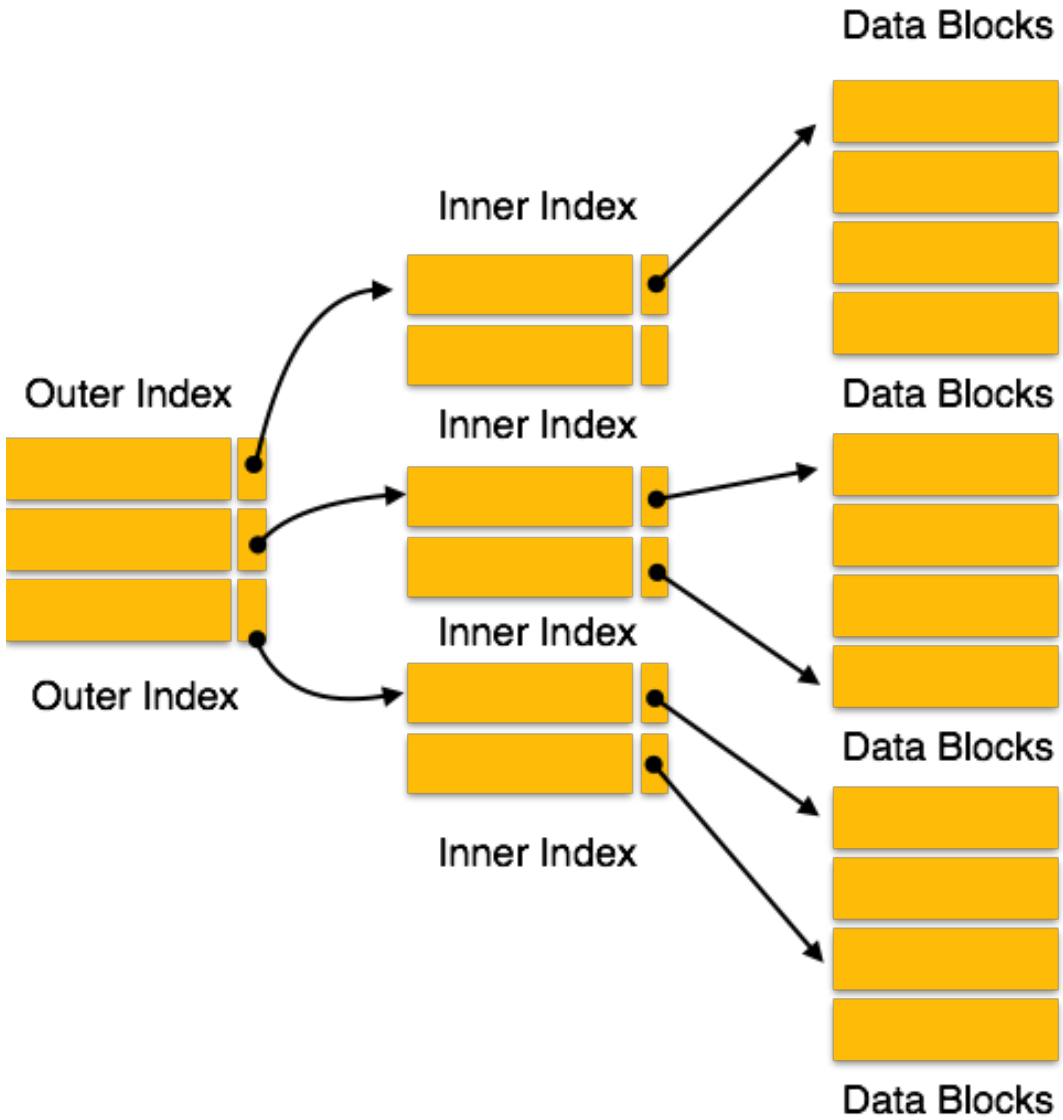
Multi-Level Indexing

Exampe-1:



Multi-Level Indexing

- Exampe-2:



Differences b/w Clustered and NonClustered Index

Parameters	Clustered	Non-clustered
Use for	You can sort the records and store clustered index physically in memory as per the order.	A non-clustered index helps you to creates a logical order for data rows and uses pointers for physical data files.
Storing method	Allows you to stores data pages in the leaf nodes of the index.	This indexing method never stores data pages in the leaf nodes of the index.
Size	The size of the clustered index is quite large.	The size of the non-clustered index is small compared to the clustered index.
Data accessing	Faster	Slower compared to the clustered index

Differences b/w Clustered and NonClustered Index

Parameters	Clustered	Non-clustered
Additional disk space	Not Required	Required to store the index separately
Type of key	By Default Primary Keys Of The Table is a Clustered Index.	It can be used with unique constraint on the table which acts as a composite key.
Main feature	A clustered index can improve the performance of data retrieval.	It should be created on columns which are used in joins.

Benefits of Clustered & Non-Clustered Index

Clustered Indexing:

- Clustered indexes are an ideal option for range or group by with max, min, count type queries
- In this type of index, a search can go straight to a specific point in data so that you can keep reading sequentially from there.
- Clustered index method uses location mechanism to locate index entry at the start of a range.
- It is an effective method for range searches when a range of search key values is requested.
- Helps you to minimize page transfers and maximize the cache hits.

Non-Clustered Indexing;

- A non-clustering index helps you to retrieves data quickly from the database table.
- Helps you to avoid the overhead cost associated with the clustered index
- A table may have multiple non-clustered indexes in RDBMS. So, it can be used to create more than one index.

Drawbacks of Clustered & Non-Clustered Index

Clustered Indexing:

- Lots of inserts in non-sequential order
- A clustered index creates lots of constant page splits, which includes data page as well as index pages.
- Extra work for SQL for inserts, updates, and deletes.
- A clustered index takes longer time to update records when the fields in the clustered index are changed.
- The leaf nodes mostly contain data pages in the clustered index.

Non-Clustered Indexing:

- A non-clustered index helps you to stores data in a logical order but does not allow to sort data rows physically.
- Lookup process on non-clustered index becomes costly.
- Every time the clustering key is updated, a corresponding update is required on the non-clustered index as it stores the clustering key.

VIT Vellore Institute of Technology

Thanks