UNIT IV - INDEX STRUCTURE, QUERY PROCESSING

PRASHANT TOMAR

Index Structures

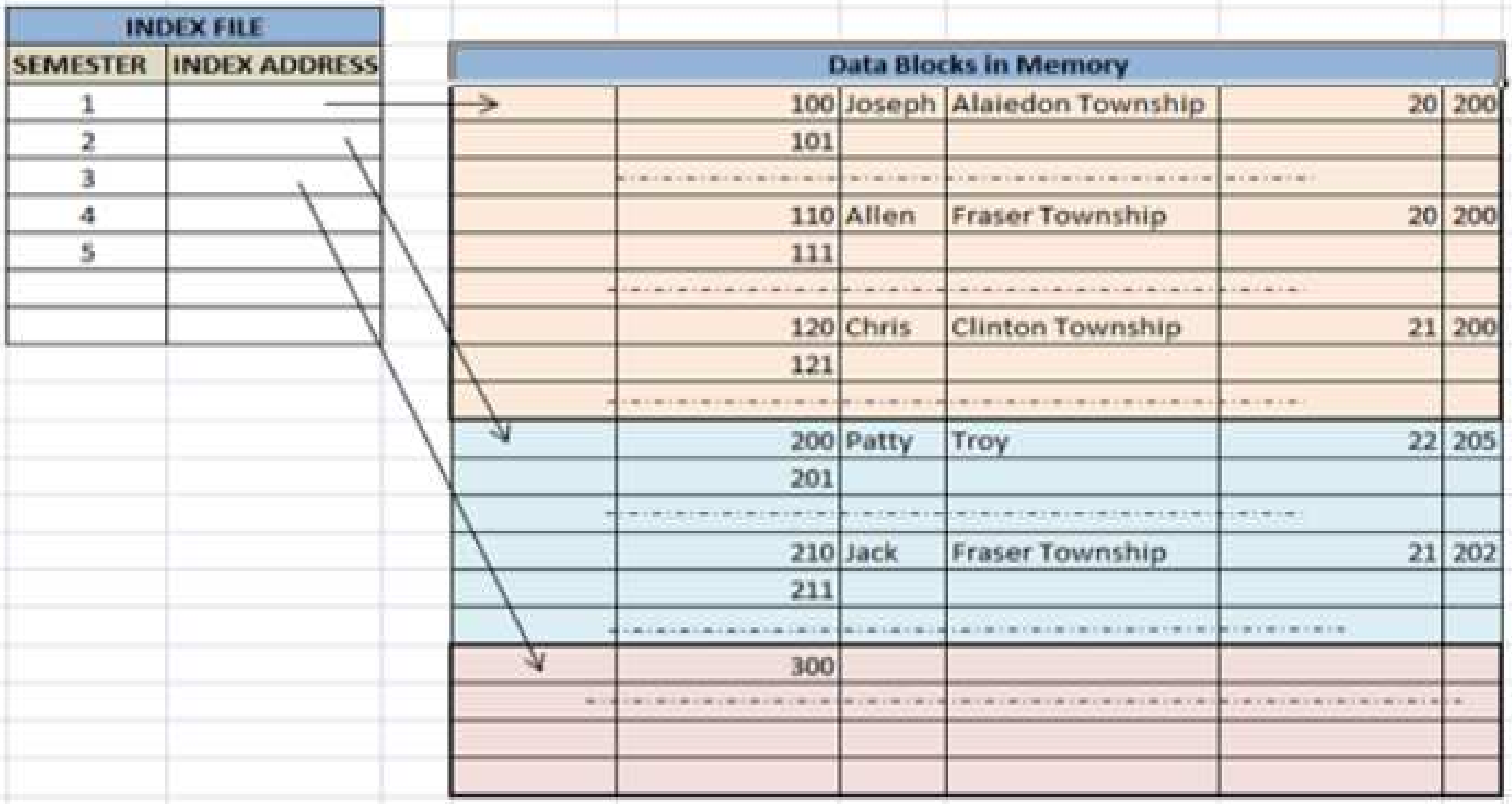
- A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure.
- Indexing is a data structure technique to efficiently retrieve records from the database files based on some attributes on which the indexing has been done. Indexing in database systems is similar to what we see in books.
- Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed.
- Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.

Index architecture/Indexing Methods

Clustered

- Clustering alters the data block into a certain distinct order to match the index, resulting in the row data being stored in order. Therefore, only one clustered index can be created on a given database table.
- Clustered indices can greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected.
- Since the physical records are in this sort order on disk, the next row item in the sequence is immediately before or after the last one, and so fewer data block reads are required.
- The primary feature of a clustered index is therefore the ordering of the physical data rows in accordance with the index blocks that point to them. Some databases separate the data and index blocks into separate files, others put two completely different data blocks within the same physical file(s).

- Clustering index is defined on an ordered data file. The data file is ordered on a non-key field. In some cases, the index is created on non-primary key columns which may not be unique for each record. In such cases, in order to identify the records faster, we will group two or more columns together to get the unique values and create index out of them. This method is known as clustering index. Basically, records with similar characteristics are grouped together and indexes are created for these groups.
- For example, students studying in each semester are grouped together. i.e. 1st Semester students, 2nd semester students, 3rdsemester students etc are grouped.



INDEX FILE	
SEMESTER	INDEX ADDRESS
1	
2	
3	
4	
5	

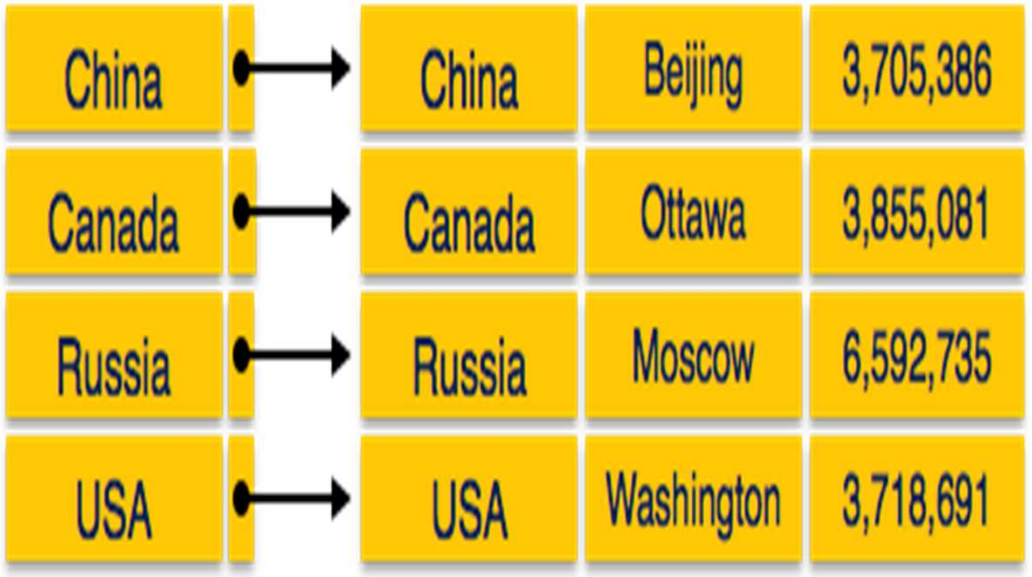
Data Blocks in Memory					
	100	Joseph	Alaiedon Township	20	200
	101				
	110	Allen	Fraser Township	20	200
	111				
	120	Chris	Clinton Township	21	200
	121				
	200	Patty	Troy	22	205
	201				
	210	Jack	Fraser Township	21	202
	211				
	300				

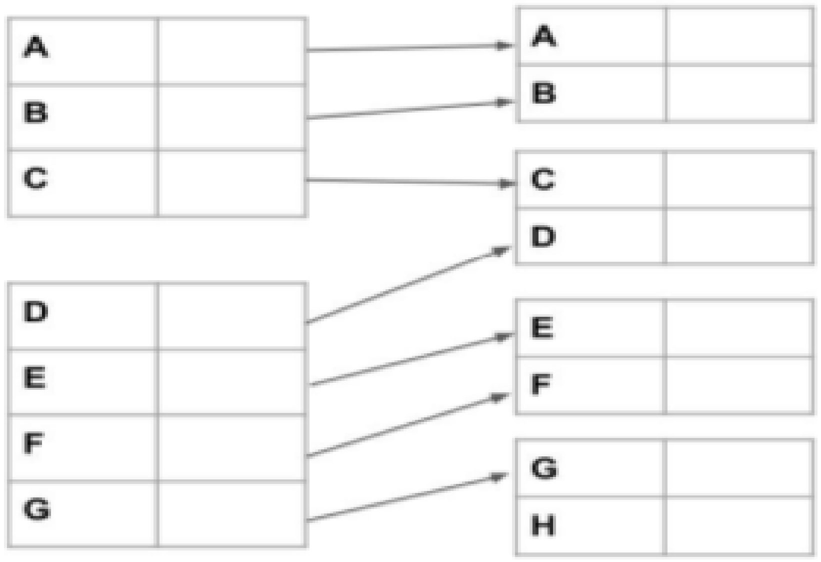
Primary Index

- In this case, the data is sorted according to the search key. It induces sequential file organisation.
- Primary index is defined on an ordered data file. The data file is ordered on a **key field**. The key field is generally the primary key of the relation.
- In this case, the primary key of the database table is used to create the index. As primary keys are unique and are stored in sorted manner, the performance of searching operation is quite efficient. The primary index is classified into two types : **Dense Index** and **Sparse Index**.

Dense Index

- In dense index, there is an index record for every search key value in the database. This makes searching faster but requires more space to store index records itself. Index records contain search key value and a pointer to the actual record on the disk.

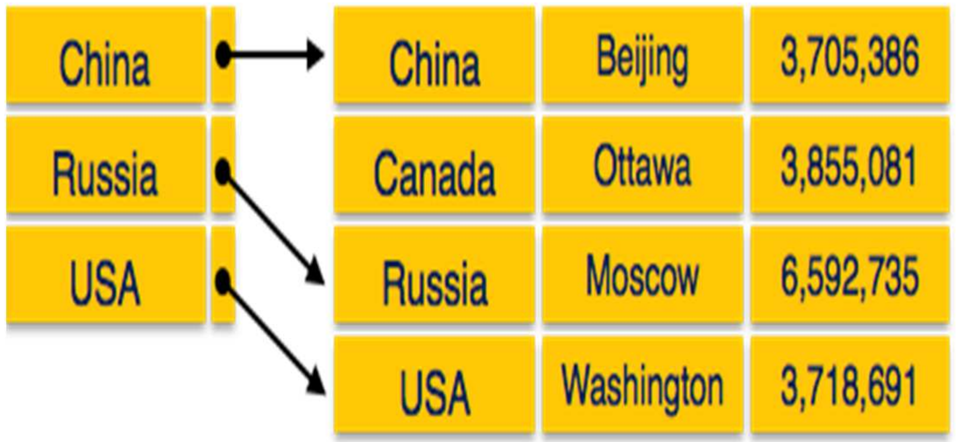


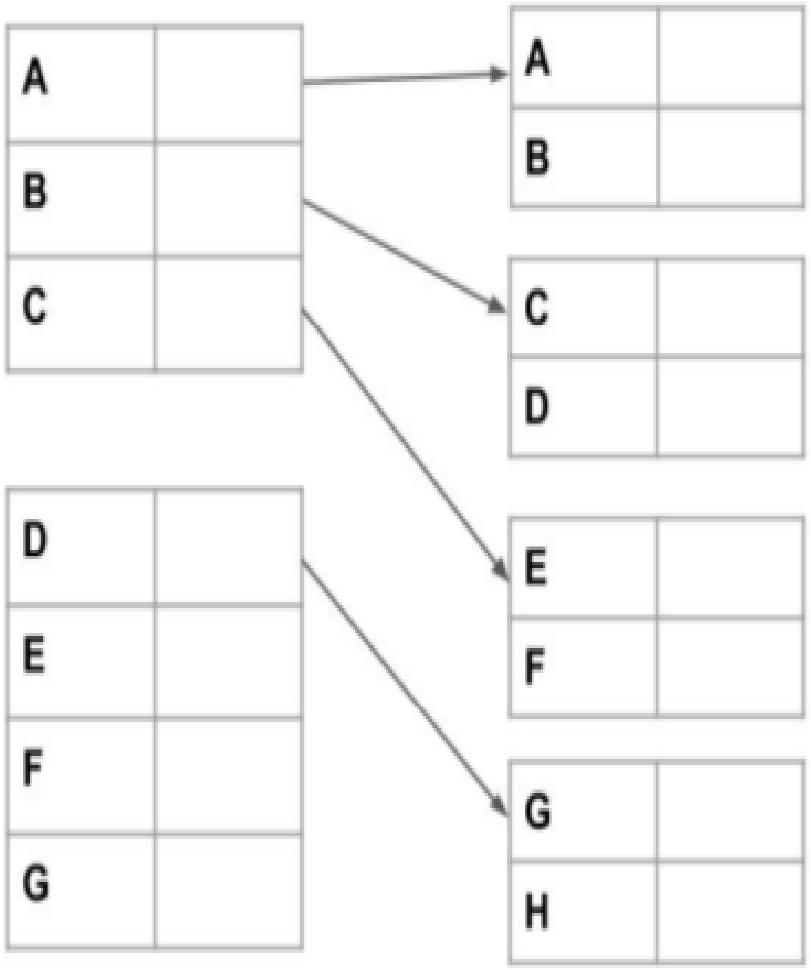


Dense Index

Sparse Index

- In sparse index, index records are not created for every search key.
- An index record here contains a search key and an actual pointer to the data on the disk.
- To search a record, we first proceed by index record and reach at the actual location of the data.
- If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.
- OR We start at that record pointed to by the index record, and proceed along the pointers in the file (that is, sequentially) until we find the desired record.





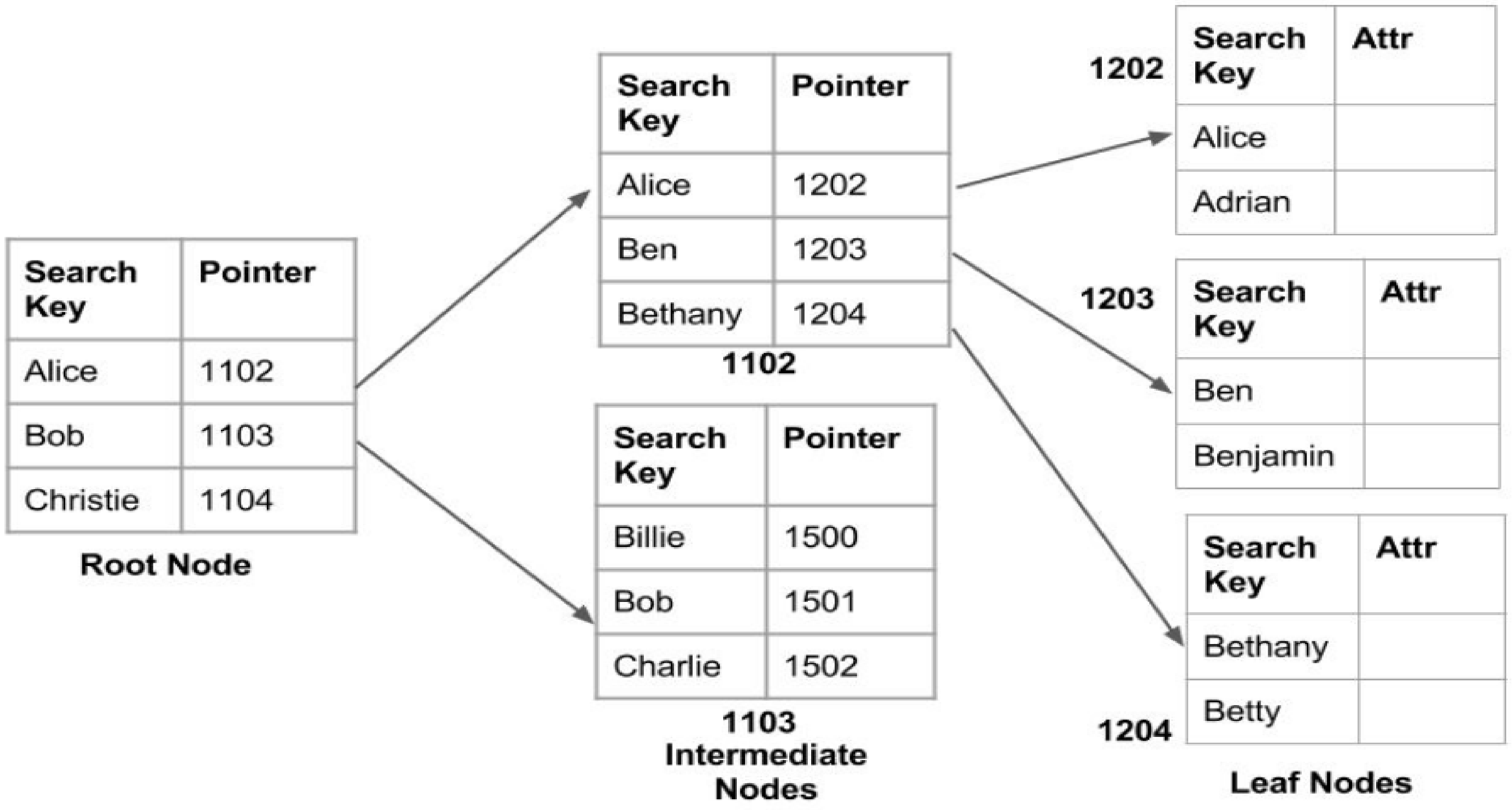
Sparse Index

Non-clustered

- The data is present in arbitrary order, but the **logical ordering** is specified by the index. The data rows may be spread throughout the table regardless of the value of the indexed column or expression.
- The non-clustered index tree contains the index keys in sorted order, with the leaf level of the index containing the pointer to the record (page and the row number in the data page in page-organized engines; row offset in file-organized engines).
- 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 book) is not organised but we have an ordered reference(contents page) to where the data points actually lie.

- In a non-clustered index,
- The physical order of the rows is not the same as the index order.
- The indexed columns are typically non-primary key columns used in JOIN, WHERE, and ORDER BY clauses.
- There can be more than one non-clustered index on a database table.



Non clustered index