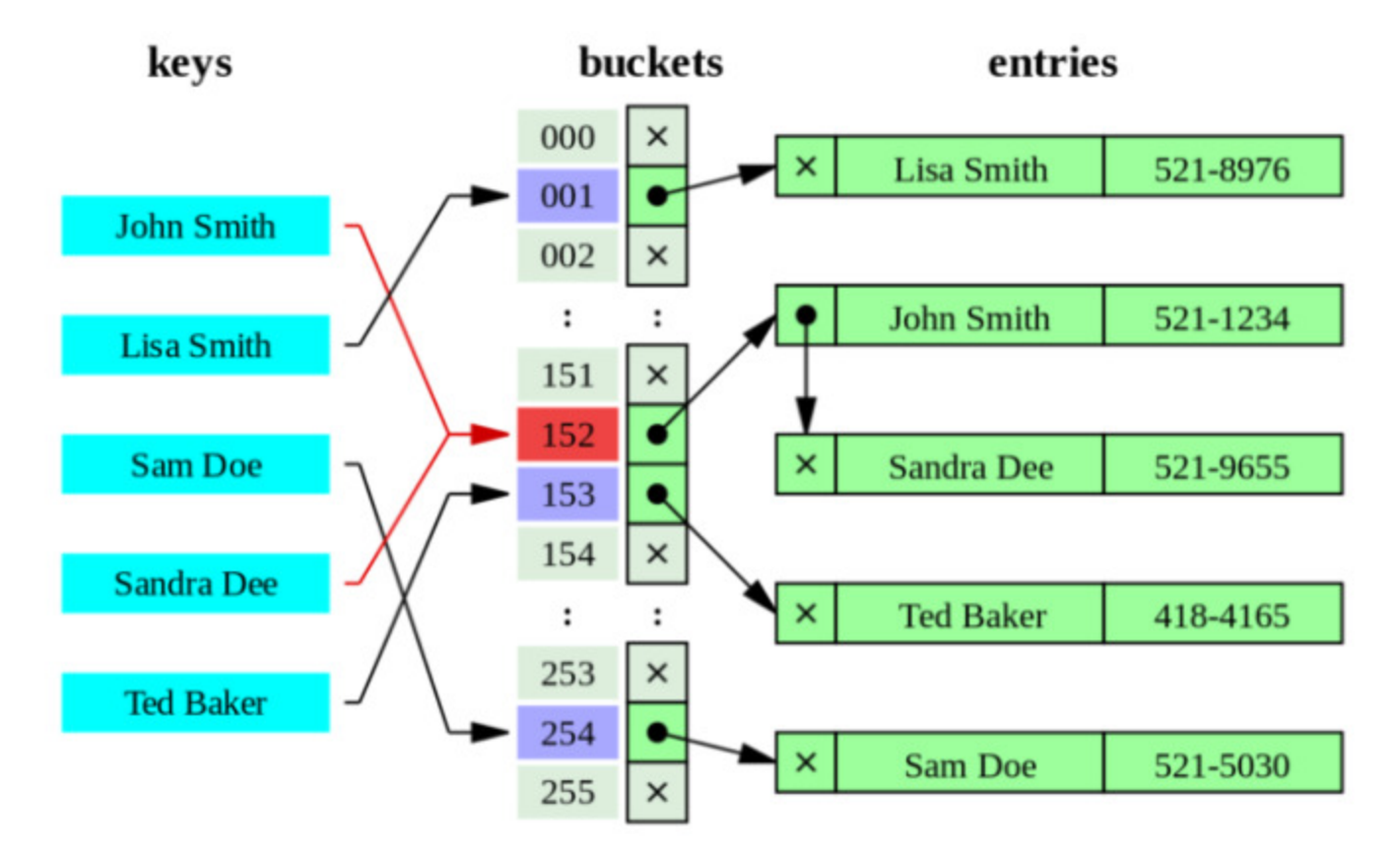
System Design — Indexes

Peng Yang Follow
Apr 6 · 6 min read



References

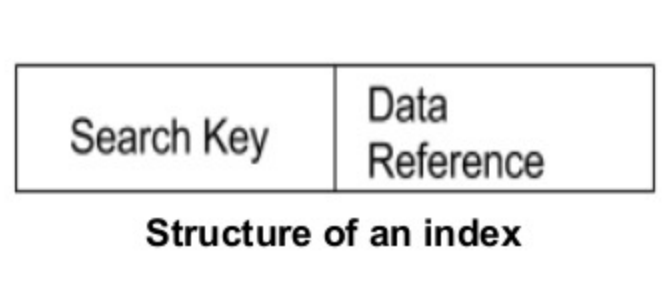
- Indexing in database
- Database indexing at a glance
- Primary and secondary index
- What is an index

Table of Contents

1. Concepts
2. Types of index
3. B-Tree
4. Advantages of Indexing
5. Disadvantages of Indexing
6. How to use index

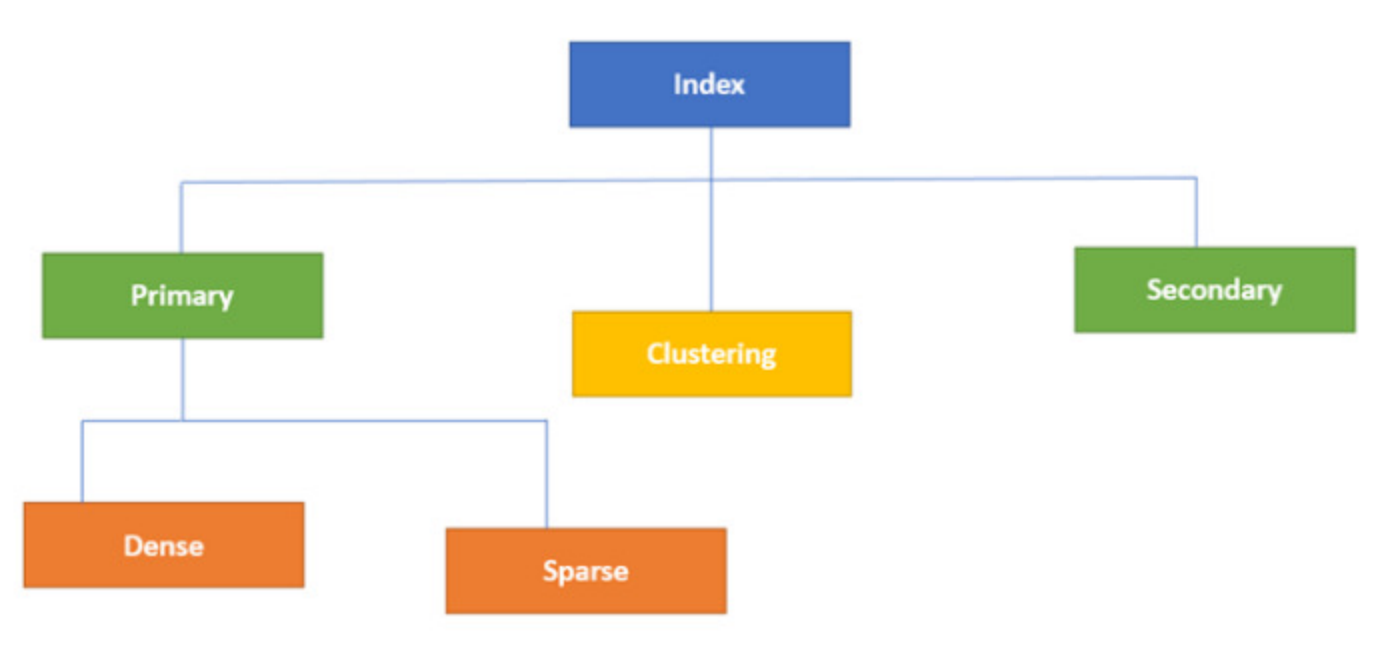
1. Concepts

- **Indexing** is a data structure technique that 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.
- Improve the performance of search queries.
- Decrease the write performance. This performance degradation applies to all insert, update, and delete operations.
- The keys are based on the tables' columns. By comparing keys to the index it is possible to find one or more database records with the same value.
- The structure that is used to store a database index is called a B+ Tree.
- Since data is constantly updated in a database, it's important for the B+ Tree to keep its balance. Each time records are added, removed, or keys updates, special algorithms shift data and key values from block to block to ensure no one part of the tree is more than one level higher than the other.
- Indexes are created using some database columns. The first column is the Search key that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly (Note that the data may or may not be stored in sorted order).
- The second column is the Data Reference which contains a set of pointers holding the address of the disk block where that particular key value can be found.



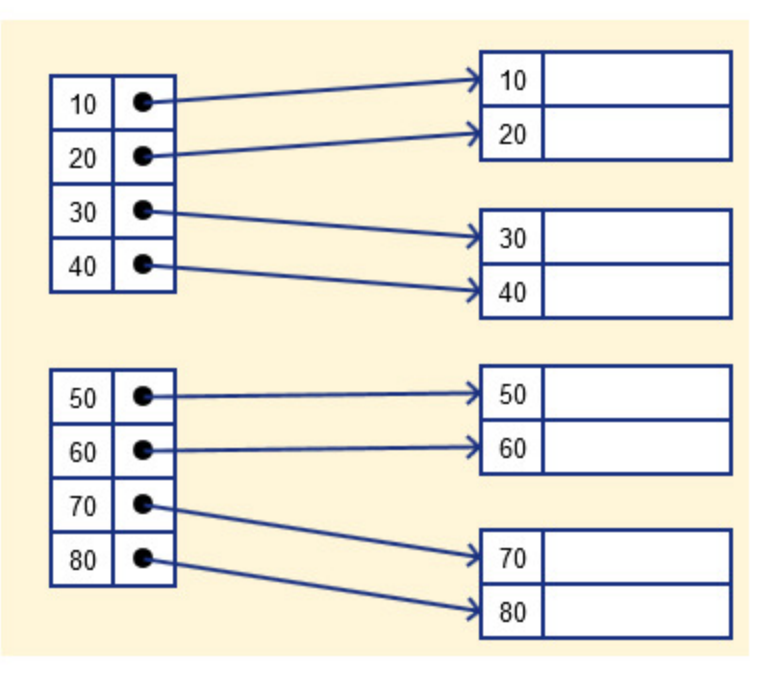
Structure of an index

2. Types of indexing methods



2.1 Primary Index

Primary Index is an ordered file which is fixed length size with two fields. The first field is the same as a primary key and the second filed 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.



Advantages of Primary Index:

- Primary index-based range queries are very efficient. There might be a possibility that the disk block that the database has read from the disk contains all the data belonging to the query since the primary index is clustered & records are ordered physically. So the locality of data can be provided by the primary index.
- Any query that takes advantage of the primary key is very fast.

Disadvantages of Primary Index:

- Since the primary index contains a direct reference to the data block address through the virtual address space & disk blocks are physically organized in the order of the index key, every time the OS does some disk page split due to `DML` operations like `INSERT` / `UPDATE` / `DELETE` , the primary index also needs to be updated. So `DML` operations put some pressure on the performance of the primary index.

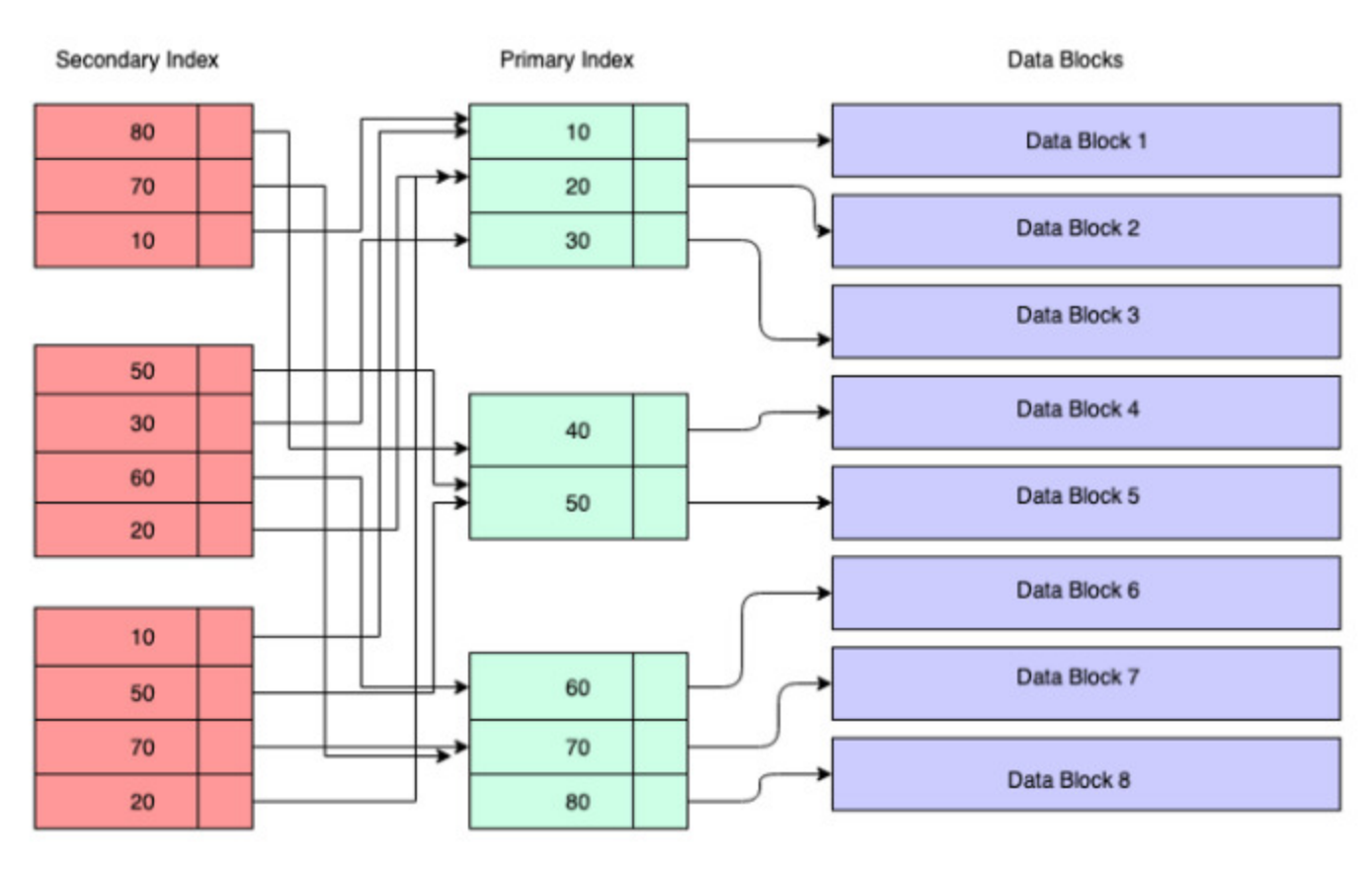
The primary Indexing is also further divided into two types:

- Dense Index
- Sparse Index

2.2 Secondary Index

The secondary Index can be generated by a field that has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index.

This two-level database indexing technique is used to reduce the mapping size of the first level. For the first level, a large range of numbers is selected because of this; the mapping size always remains small.



Advantages of a Secondary Index:

Logically you can create as many secondary indices as you want. But in reality how many indices actually required needs a serious thought process since each index has its own penalty.

Disadvantages of a Secondary Index:

With `DML` operations like `DELETE` / `INSERT` , the secondary index also needs to be updated so that the copy of the primary key column can be deleted / inserted. In such cases, the existence of lots of secondary indexes can create issues.

2.3 Clustering Index

Records themselves are stored in the Index and not pointers. Sometimes the Index is created on non-primary key columns which might not be unique for each record. In such a situation, you can group two or more columns to get the unique values and create an index which is called clustered Index. This also helps you to identify the record faster.

Advantage of Clustered Index:

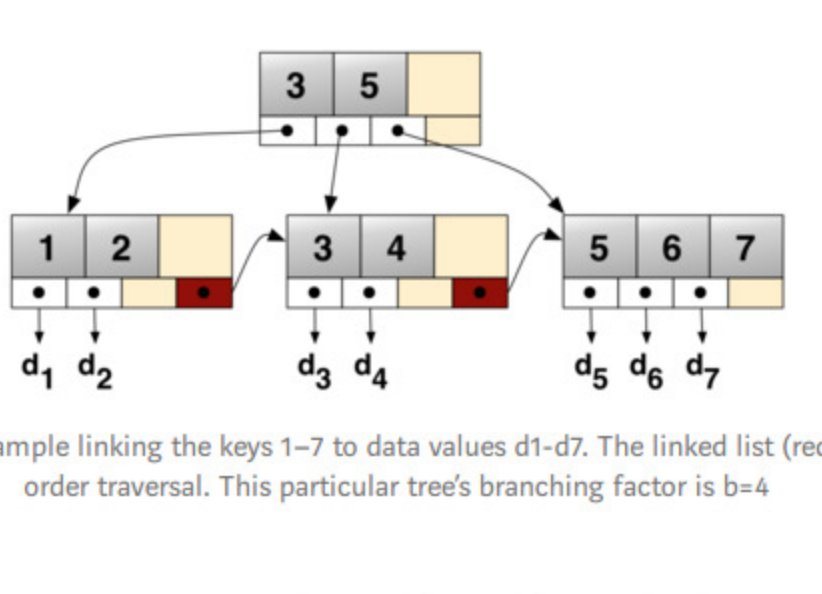
This ordering or co-location of related data actually makes a clustered index faster. When data is fetched from disk, the complete block containing the data is read by the system since our disk IO system writes & reads data in blocks. So in case of range queries, it's quite possible that the collocated data is buffered in memory.

Constraints of Clustered Index:

Since a clustered index impacts the physical organization of the data, there can be only one clustered index per table.

3. B-Tree Index

B-tree index is the widely used data structures for Indexing. It is a multilevel index format technique which is balanced binary search trees. All leaf nodes of the B tree signify actual data pointers. While the advantages of the B-Tree are numerous, the main advantage for our purposes is that it is sortable. When the data structure is sorted in order, it makes our search more efficient for the obvious reasons we pointed out above. When the index creates a data structure on a specific column, it is important to note that no other column is stored in the data structure.



A simple B+ tree example linking the keys 1–7 to data values d1-d7. The linked list (red) allows rapid in-order traversal. This particular tree's branching factor is b=4

Some possible candidates are — `BTREE` , `HASH` , `RTREE` or `FULLTEXT` .

4. Advantages of Indexing

Important pros/ advantage of Indexing are:

- It helps you to reduce the total number of I/O operations needed to retrieve that data, so you don't need to access a row in the database from an index structure.
- Offers Faster search and retrieval of data to users.
- Indexing also helps you to reduce tablespace as you don't need to link to a row in a table, as there is no need to store the ROWID in the Index. Thus you will able to reduce the tablespace.
- You can't sort data in the lead nodes as the value of the primary key classifies it.

5. Disadvantages of Indexing

Important drawbacks/cons of Indexing are:

- To perform the indexing database management system, you need a primary key on the table with a unique value.
- You can't perform any other indexes on the Indexed data.
- You are not allowed to partition an index-organized table.
- SQL Indexing Decrease performance in INSERT, DELETE, and UPDATE query.

6. What Clauses use index?

It is a common misconception that indexes only help the `where` clause. B-tree indexes can also help the `order by` , `group by` , `select` and other clauses. It is just the B-tree part of an index—not the doubly linked list—that cannot be used by other clauses.

Thank you for reading! if you liked this blog, please also check my other blogs of System Design series.

- System Design — Load Balancing
- System Design — Caching
- System Design — Sharding / Data Partitioning
- System Design — Indexes
- System Design — Proxies
- System Design — Message Queues
- System Design — Redundancy and Replication
- System Design — SQL vs. NoSQL
- System Design — CAP Problem
- System Design — Consistent Hashing
- System Design — Client-Server Communication
- System Design — Storage
- System Design — Other Topics
- Object-Oriented Programming — Basic Design Patterns in C++

System Design Interview · Index

WRITTEN BY
Peng Yang Follow
Software/Infrastructure Engineer. Aim to become an engineer who understands computer science very well. https://www.linkedin.com/in/peng-larry-yang-9a794561/

Computer Science Fundamentals Follow
Computer Science Fundamentals