

Database Indexes Simplified

Let's say we have a table or series of documents in a database with the following details.

ID	FirstName	LastName
1	Troy	Vaughn
2	James	Gray
3	Sally	Smith
5	William	Berkley
7	Douglas	Adams
10	Patty	Mills

Note: *I have used a table to represent my data in this example, but the same concepts would apply to a collection in Mongo DB, it would just look slightly different.*

When created, this set of records will already get an index for the ID field which is a common search parameter for queries. In relational databases this is easy to setup because identities are always unique and already sequential due to auto incrementing. This will make them fast to search on because you can use a binary search or similar, to find the ID you are after faster than checking each one in sequence.

Now let's say we want to make the LastName field better for querying, in this case they are unlikely to already be in order so if we want to search for a particular last name, the database will need check each one sequentially until it finds the one it is after.

However, if we create an index for the LastName field, we can make this faster. When we create an index for the LastName field a structure will be created that stores all the last names in order alongside the address/location of the full entry in the original table/collection. This can be done either by some kind of memory address pointer, row number value, or other identifier.

LAST NAME	Pointer/Address
Adams	_7
Berkley	_5
Gray	_2
Mills	_10
Smith	_3
Vaughn	_1

NOTE: Although The image above shows the details of the indexes in a table, this is only done to simplify this example. In most databases this would really be stored in some type of tree structure such as a B+ Tree.

Now when we search by last name, instead of checking each name sequentially, it can do a binary search or similar searching algorithm to find the name it is looking for from within the index structure. Then once it is found it can use the location address stored with the name to go to the main table/collection and retrieve the desired record.

LAST NAME	Pointer/Address
Adams	_7
Berkley	_5
Gray	_2
Mills	_10
Smith	_3
Vaughn	_1

ID	FirstName	LastName
1	Troy	Vaughn
2	James	Gray
3	Sally	Smith
5	William	Berkley
7	Douglas	Adams
10	Patty	Mills