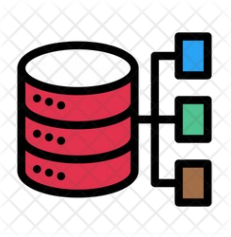
MongoDB Internals

Tushar B. Kute,
http://tusharkute.com



Index

- An index is a data structure that allows us to add indexes in the existing table. It enables you to improve the faster retrieval of records on a database table.
- It creates an entry for each value of the indexed columns. We use it to quickly find the record without searching each row in a database table whenever the table is accessed.
- We can create an index by using one or more columns of the table for efficient access to the records.

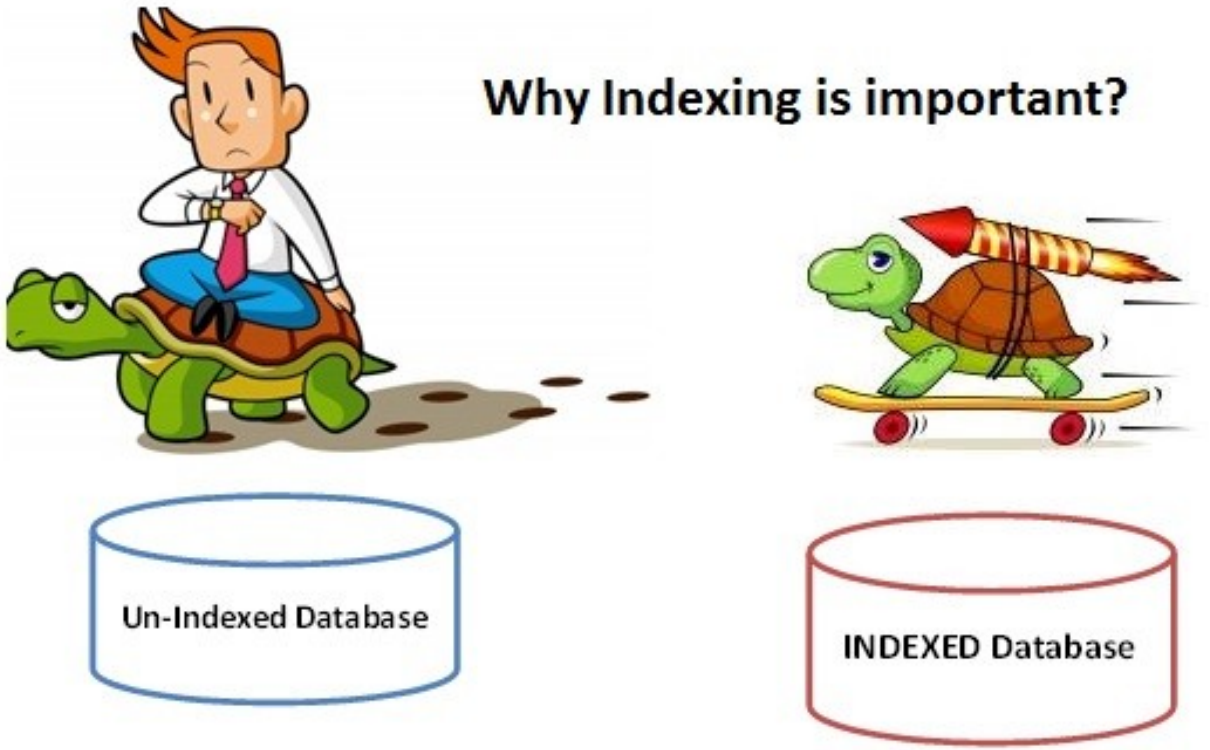
Index

- When a table is created with a primary key or unique key, it automatically creates a special index named PRIMARY.
- We called this index as a clustered index. All indexes other than PRIMARY indexes are known as a non-clustered index or secondary index.

Indexing need

- Suppose we have a contact book that contains names and mobile numbers of the user. In this contact book, we want to find the mobile number of Martin Williamson.
- If the contact book is an unordered format means the name of the contact book is not sorted alphabetically, we need to go over all pages and read every name until we will not find the desired name that we are looking for.
- This type of searching name is known as sequential searching.

Why indexing ?



Index

- Indexes support the efficient resolution of queries. Without indexes, MongoDB must scan every document of a collection to select those documents that match the query statement.
- This scan is highly inefficient and require MongoDB to process a large volume of data.
- Indexes are special data structures, that store a small portion of the data set in an easy-to-traverse form.
- The index stores the value of a specific field or set of fields, ordered by the value of the field as specified in the index.

Index

- To create an index, you need to use createIndex() method of MongoDB.
- Syntax:

  The basic syntax of createIndex() method is as follows().

  > db.COLLECTION_NAME.createIndex({KEY:1})
- Here key is the name of the field on which you want to create index and 1 is for ascending order.

Index

- To create index in descending order you need to use -1.
- Example

```
>db.mycol.createIndex({"title":1})
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 1,
"numIndexesAfter" : 2,
"ok" : 1
}
```

Index

- In createIndex() method you can pass multiple fields, to create index on multiple fields.

```
>db.mycol.createIndex({"title":1,"description":-1})
```

Index

- You can drop a particular index using the dropIndex() method of MongoDB.
- Syntax:

  > db.COLLECTION_NAME.dropIndex({KEY:1})

- Here, "key" is the name of the file on which you want to remove an existing index. Instead of the index specification document (above syntax), you can also specify the name of the index directly as:

  dropIndex("name_of_the_index")

Index

- Example:

```
> db.mycol.dropIndex({"title":1})
{
"ok" : 0,
"errmsg" : "can't find index with key: { title:
1.0 }",
"code" : 27,
"codeName" : "IndexNotFound"
}
```

Index

- This method deletes multiple (specified) indexes on a collection.
- Syntax:

  > db.COLLECTION_NAME.dropIndexes()
- Example
- Assume we have created 2 indexes in the named mycol collection as shown below –

  > db.mycol.createIndex({"title":1,"description":-1})
- Following example removes the above created indexes of mycol –

  > db.mycol.dropIndexes({"title":1,"description":-1})

  { "nIndexesWas" : 2, "ok" : 1 }

Index

- The getIndexes() method

- This method returns the description of all the indexes int the collection.

  Syntax:

  ```
  db.COLLECTION_NAME.getIndexes( )
  ```

Thank you

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com