**Indexes** support the efficient execution of queries in MongoDB.

Without indexes, MongoDB must perform a collection scan, i.e. scan every document in a collection, to select those documents that match the query statement. If an appropriate index exists for a query, MongoDB can use the index to limit the number of documents it must inspect.

Indexes are special data structures that store a small portion of the collection’s 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.

The ordering of the index entries supports efficient equality matches and range-based query operations. In addition, MongoDB can return sorted results by using the ordering in the index.

Fundamentally, indexes in MongoDB are similar to indexes in other database systems. MongoDB defines indexes at the collection level and supports indexes on any field or sub-field of the documents in a MongoDB collection.

MongoDB creates a unique index on the \_id field during the creation of a collection. The \_id index prevents clients from inserting two documents with the same value for the \_id field. You cannot drop this index on the \_id field.

To create an index in the Mongo Shell,

use db.collection.createIndex()

MongoDB indexes use a B-tree data structure

**Index Names**

The default name for an index is the concatenation of the indexed keys and each key’s direction in the index ( i.e. 1 or -1) using underscores as a separator. For example, an index created on { item : 1, quantity: -1 } has the name item\_1\_quantity\_-1.

db.student.createIndex( {District:1 , Student\_Name:1})

You can view index names using the db.collection.getIndexes() method. You cannot rename an index once created. Instead, you must drop and re-create the index with a new name.