

Practical 6 : Indexing In MongoDB

-Indexes can be created using the `createIndex()` method or by specifying indexes in a document's schema for collections using MongoDB's schema validation feature.

-Suppose the document is like

```
{
  _id: ObjectId('66104eb5704ada6a58271bda'),
  name: 'Everest',
  height: 8848,
  location: [ 'Nepal', 'China' ],
  ascents: {
    first: { year: 1953 },
    first_winter: { year: 1980 },
    total: 5656
  }
}
```

```
db.peaks.find(
  { "height": { $gt: 8700 } }
).explain("executionStats")
This operation returns a lot of information.
```

- Creating a Single Field Index

```
db.peaks.createIndex( { "height": 1 } )
height_1
```

-Retrieve the list of created indexes

```
student> db.peaks.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { height: 1 }, name: 'height_1' }
]
```

-Find documents where height greater than 8700

```
db.peaks.find(
  { "height": { $gt: 8700 } }
).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'student.peaks',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'Everest' } },
    queryHash: 'A2F868FD',
    planCacheKey: 'A2F868FD',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: { name: { '$eq': 'Everest' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 5,
  }
}
```

```

        executionStages: {
          stage: 'COLLSCAN',
          filter: { name: { '$eq': 'Everest' } } },
          nReturned: 1,
          executionTimeMillisEstimate: 0,
          works: 6,
          advanced: 1,
          needTime: 4,
          needYield: 0,
          saveState: 0,
          restoreState: 0,
          isEOF: 1,
          direction: 'forward',
          docsExamined: 5
        }
      },
      command: { find: 'peaks', filter: { name: 'Everest' }, '$db':
'student' },
      serverInfo: {
        host: 'LAPTOP-1C3IMV0U',
        port: 27017,
        version: '7.0.5',
        gitVersion: '7809d71e84e314b497f282ea8aa06d7ded3eb205'
      },
      serverParameters: {
        internalQueryFacetBufferSizeBytes: 104857600,
        internalQueryFacetMaxOutputDocSizeBytes: 104857600,
        internalLookupStageIntermediateDocumentMaxSizeBytes:
104857600,
        internalDocumentSourceGroupMaxMemoryBytes: 104857600,
        internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
        internalQueryProhibitBlockingMergeOnMongoS: 0,
        internalQueryMaxAddToSetBytes: 104857600,
        internalDocumentSourceSetWindowFieldsMaxMemoryBytes:
104857600,
        internalQueryFrameworkControl: 'trySbeRestricted'
      },
      ok: 1
    }
  }
}

```

```

-Deleting the index
db.peaks.dropIndex('height_1')
{ nIndexesWas: 2, ok: 1 }

```

```

student> student> db.peaks.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
student>

```