

Client Side:

```
student@dbms-8:~/Desktop/MongoDB/mongo/bin$ ./mongo
MongoDB shell version: 2.6.3
connecting to: test
> use mmcoe;
switched to db mmcoe
```

Query to insert multiple documents by using for loop:

```
>
for(i=5;i<10;i++){db.staff.insert({tid:i,tname:'teacher'+i,age:Math.floor(Math.random()*120),
doj:new Date()});
WriteResult({ "nInserted" : 1 })

> db.staff.find();
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24c"), "tid" : 5, "tname" : "teacher5", "age" : 53,
"doj" : ISODate("2015-09-04T05:55:11.124Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24d"), "tid" : 6, "tname" : "teacher6", "age" : 86,
"doj" : ISODate("2015-09-04T05:55:11.163Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24e"), "tid" : 7, "tname" : "teacher7", "age" : 64,
"doj" : ISODate("2015-09-04T05:55:11.165Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24f"), "tid" : 8, "tname" : "teacher8", "age" : 102,
"doj" : ISODate("2015-09-04T05:55:11.166Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b250"), "tid" : 9, "tname" : "teacher9", "age" : 74,
"doj" : ISODate("2015-09-04T05:55:11.167Z") }
```

Query to insert multiple documents by using for loop:

```
>
for(i=1;i<5;i++){db.staff.insert({tid:i,tname:'teacher'+i,age:Math.floor(Math.random()*120),
doj:new Date()});
WriteResult({ "nInserted" : 1 })
> db.staff.find();
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24c"), "tid" : 5, "tname" : "teacher5", "age" : 53,
"doj" : ISODate("2015-09-04T05:55:11.124Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24d"), "tid" : 6, "tname" : "teacher6", "age" : 86,
"doj" : ISODate("2015-09-04T05:55:11.163Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24e"), "tid" : 7, "tname" : "teacher7", "age" : 64,
"doj" : ISODate("2015-09-04T05:55:11.165Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24f"), "tid" : 8, "tname" : "teacher8", "age" : 102,
"doj" : ISODate("2015-09-04T05:55:11.166Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b250"), "tid" : 9, "tname" : "teacher9", "age" : 74,
"doj" : ISODate("2015-09-04T05:55:11.167Z") }
{ "_id" : ObjectId("55e9326408de6a2c6a43b251"), "tid" : 1, "tname" : "teacher1", "age" : 8,
"doj" : ISODate("2015-09-04T05:55:48.566Z") }
{ "_id" : ObjectId("55e9326408de6a2c6a43b252"), "tid" : 2, "tname" : "teacher2", "age" :
108, "doj" : ISODate("2015-09-04T05:55:48.567Z") }
{ "_id" : ObjectId("55e9326408de6a2c6a43b253"), "tid" : 3, "tname" : "teacher3", "age" : 40,
"doj" : ISODate("2015-09-04T05:55:48.569Z") }
```

```
{ "_id" : ObjectId("55e9326408de6a2c6a43b254"), "tid" : 4, "tname" : "teacher4", "age" : 57,
"doj" : ISODate("2015-09-04T05:55:48.569Z") }
```

Query to save the documents:

```
> db.staff.save({});
WriteResult({ "nInserted" : 1 })
```

Query to find the documents where teacher name is teacher1 and get the additional information regarding the query by using .explain() method

```
> db.staff.find({tname:'teacher1'}).explain();
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 1,
  "nscannedObjects" : 10,
  "nscanned" : 10,
  "nscannedObjectsAllPlans" : 10,
  "nscannedAllPlans" : 10,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 1,
  "nChunkSkips" : 0,
  "millis" : 16,
  "server" : "dbms-8:27017",
  "filterSet" : false
}
```

Query to create the index on the key tname

```
> db.staff.ensureIndex({tname:1});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Query to check the created indexes on the collection

```
> db.staff.getIndexes();
[
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "mmcoe.staff"
```

```

    },
    {
        "v" : 1,
        "key" : {
            "tname" : 1
        },
        "name" : "tname_1",
        "ns" : "mmcoe.staff"
    }
]

```

Querying again the same teacher document after creating the index on tname

//significant differences in nscanned objects, millis, etc., changed the cursor from basic cursor to B_tree_cursor

```

> db.staff.find({ tname:'teacher1'}).explain();
{
  "cursor" : "BtreeCursor tname_1",
  "isMultiKey" : false,
  "n" : 1,
  "nscannedObjects" : 1,
  "nscanned" : 1,
  "nscannedObjectsAllPlans" : 1,
  "nscannedAllPlans" : 1,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 15,
  "indexBounds" : {
    "tname" : [
      [
        "teacher1",
        "teacher1"
      ]
    ]
  },
  "server" : "dbms-8:27017",
  "filterSet" : false
}

```

Query to find the documents where age is 40 & tname for the qualifying condition

//It still uses the index created on tname

```

> db.staff.find({ age:40,tname: {'$gt':'teacher5', '$lt':'teacher9'}}).explain();
{
  "cursor" : "BtreeCursor tname_1",
  "isMultiKey" : false,

```

```

    "n" : 0,
    "nscannedObjects" : 3,
    "nscanned" : 3,
    "nscannedObjectsAllPlans" : 3,
    "nscannedAllPlans" : 3,
    "scanAndOrder" : false,
    "indexOnly" : false,
    "nYields" : 0,
    "nChunkSkips" : 0,
    "millis" : 0,
    "indexBounds" : {
      "tname" : [
        [
          "teacher5",
          "teacher9"
        ]
      ]
    },
    "server" : "dbms-8:27017",
    "filterSet" : false
  }
}

```

Query to create compound index on age & tname both

```

> db.staff.ensureIndex({age:1,tname:1});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 2,
  "numIndexesAfter" : 3,
  "ok" : 1
}
> db.staff.getIndexes();
[
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "mmcoe.staff"
  },
  {
    "v" : 1,
    "key" : {
      "tname" : 1
    },
    "name" : "tname_1",
    "ns" : "mmcoe.staff"
  },
  {

```

```

        "v" : 1,
        "key" : {
            "age" : 1,
            "tname" : 1
        },
        "name" : "age_1_tname_1",
        "ns" : "mmcoe.staff"
    }
]

```

Query as above to find the age:40 & tname now uses compound index

```

> db.staff.find({ age:40,tname: {'$gt':'teacher5', '$lt':'teacher9'}}).explain();
{
  "cursor" : "BtreeCursor age_1_tname_1",
  "isMultiKey" : false,
  "n" : 0,
  "nscannedObjects" : 0,
  "nscanned" : 0,
  "nscannedObjectsAllPlans" : 1,
  "nscannedAllPlans" : 2,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "indexBounds" : {
    "age" : [
      [
        40,
        40
      ]
    ],
    "tname" : [
      [
        "teacher5",
        "teacher9"
      ]
    ]
  },
  "server" : "dbms-8:27017",
  "filterSet" : false
}

```

Another query to use the compound index:

```

> db.staff.find({ age: {'$gt':10},tname: {'$gt':'teacher5', '$lt':'teacher9'}});

```

```
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24d"), "tid" : 6, "tname" : "teacher6", "age" : 86,
"doj" : ISODate("2015-09-04T05:55:11.163Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24e"), "tid" : 7, "tname" : "teacher7", "age" : 64,
"doj" : ISODate("2015-09-04T05:55:11.165Z") }
{ "_id" : ObjectId("55e9323f08de6a2c6a43b24f"), "tid" : 8, "tname" : "teacher8", "age" : 102,
"doj" : ISODate("2015-09-04T05:55:11.166Z") }
```

```
> db.staff.find({age: {'$gt':10}, tname: {'$gt': 'teacher5', '$lt': 'teacher9'}}).explain();
{
  "cursor" : "BtreeCursor tname_1",
  "isMultiKey" : false,
  "n" : 3,
  "nscannedObjects" : 3,
  "nscanned" : 3,
  "nscannedObjectsAllPlans" : 6,
  "nscannedAllPlans" : 11,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "indexBounds" : {
    "tname" : [
      [
        "teacher5",
        "teacher9"
      ]
    ]
  },
  "server" : "dbms-8:27017",
  "filterSet" : false
}
```

Query to drop the index:

```
> db.staff.dropIndex('age_1_tname_1');
{ "nIndexesWas" : 3, "ok" : 1 }
```

After dropping the compound index it again uses index created on t_name

```
> db.staff.find({age: {'$gt':10}, tname: {'$gt': 'teacher5', '$lt': 'teacher9'}}).explain();
{
  "cursor" : "BtreeCursor tname_1",
  "isMultiKey" : false,
  "n" : 3,
  "nscannedObjects" : 3,
  "nscanned" : 3,
  "nscannedObjectsAllPlans" : 3,
  "nscannedAllPlans" : 3,
```

```

    "scanAndOrder" : false,
    "indexOnly" : false,
    "nYields" : 0,
    "nChunkSkips" : 0,
    "millis" : 0,
    "indexBounds" : {
      "tname" : [
        [
          "teacher5",
          "teacher9"
        ]
      ]
    },
    "server" : "dbms-8:27017",
    "filterSet" : false
  }
}

```

Query to drop the index created on tname

```

> db.staff.dropIndex('tname_1');
{ "nIndexesWas" : 2, "ok" : 1 }

```

//Result again uses the basic cursor, this time nscanned objects are 10

```

> db.staff.find({ age: {'$gt':10}, tname: {'$gt':'teacher5', '$lt':'teacher9'}}).explain();
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 3,
  "nscannedObjects" : 10,
  "nscanned" : 10,
  "nscannedObjectsAllPlans" : 10,
  "nscannedAllPlans" : 10,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 0,
  "nChunkSkips" : 0,
  "millis" : 0,
  "server" : "dbms-8:27017",
  "filterSet" : false
}

```

Query to create Unique index

```

> db.staff.ensureIndex({ tname:1 }, { unique:1 });
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,

```

```

        "ok" : 1
    }
    > db.staff.getIndexes();
    [
      {
        "v" : 1,
        "key" : {
          "_id" : 1
        },
        "name" : "_id_",
        "ns" : "mmcoe.staff"
      },
      {
        "v" : 1,
        "unique" : true,
        "key" : {
          "tname" : 1
        },
        "name" : "tname_1",
        "ns" : "mmcoe.staff"
      }
    ]

```

Query to insert the document with duplicate values on which unique index is created

//Error is generated as per the functionality

```

> db.staff.insert({"tid" : 6, "tname" : "teacher6", "age" : 86});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error
index: mmcoe.staff.$tname_1  dup key: { : \"teacher6\" }"
  }
})
>

```