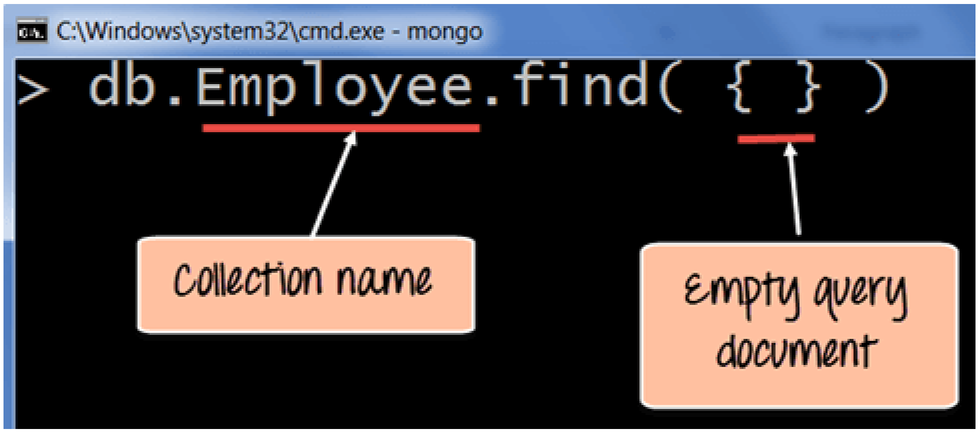
MongoDB Query Document using find ()

- The method of fetching or getting data from a MongoDB database is carried out by using queries. While performing a query operation, one can also use criteria's or conditions which can be used to retrieve specific data from the database?
- MongoDB provides a function called **db.collection.find ()** which is used for retrieval of documents from a MongoDB database.

Basic query operations

The basic query operations cover the simple operations such as getting all of the documents in a MongoDB collection. Let's look at an example of how we can accomplish this.

All of our code will be run in the MongoDB JavaScript command shell. Consider that we have a collection named 'Employee' in our MongoDB database and we execute the below command.



```
> db.Employee.find( { } )
```

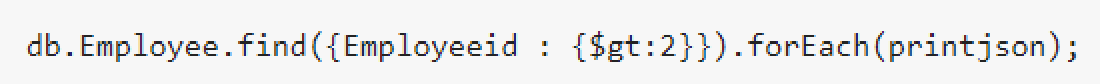
Example 1

Let's look at a couple of examples of how we can accomplish this.

```
db.Employee.find({EmployeeName : "Smith"}).forEach(printjson);
```

Example 2

Now, let's take a look at another code example which makes use of the greater than search criteria. When this criterion is included, it actually searches those documents where the value of the field is greater than the specified value.



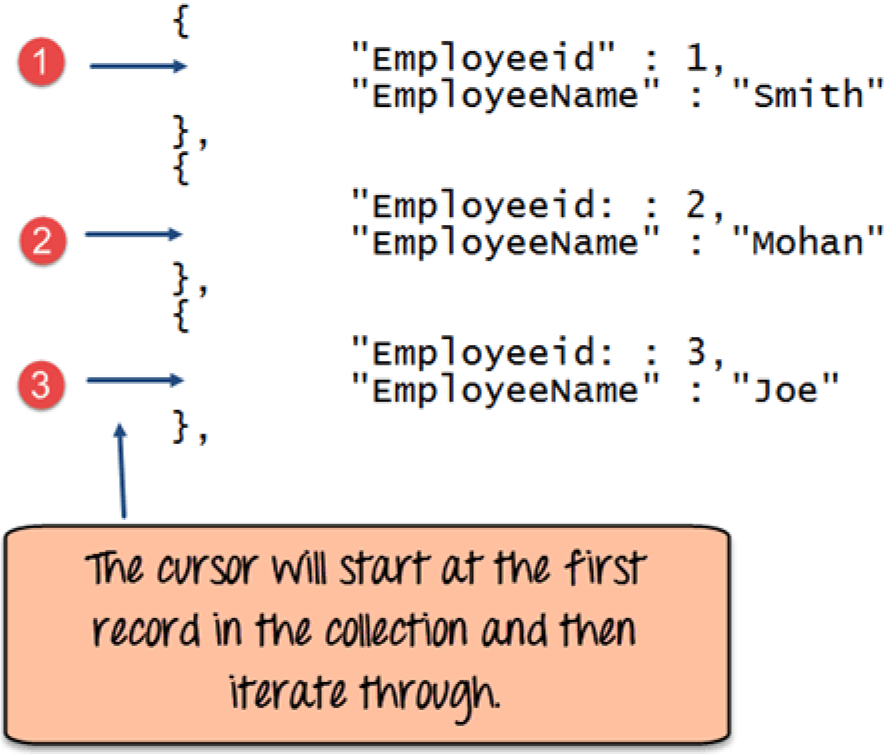
```
db.Employee.find({Employeeid : {$gt:2}}).forEach(printjson);
```

MongoDB Cursor

What is Cursor in MongoDB?

When the **db.collection.find ()** function is used to search for documents in the collection, the result returns a pointer to the collection of documents returned which is called a cursor.

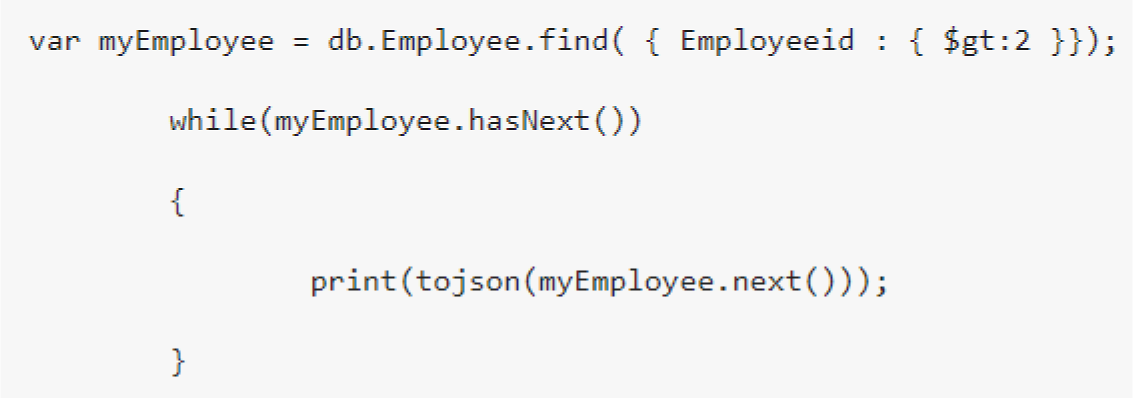
By default, the cursor will be iterated automatically when the result of the query is returned. But one can also explicitly go through the items returned in the cursor one by one. If you see the below example, if we have 3 documents in our collection, the cursor object will point to the first document and then iterate through all of the documents of the collection.



```
{
        "Employeeid" : 1,
        "EmployeeName" : "Smith"
},
{
        "Employeeid: : 2,
        "EmployeeName" : "Mohan"
},
{
        "Employeeid: : 3,
        "EmployeeName" : "Joe"
},
```

The cursor will start at the first record in the collection and then iterate through.

The following example shows how this can be done.



```
var myEmployee = db.Employee.find( { Employeeid : { $gt:2 }});

        while(myEmployee.hasNext())

        {

                print(tojson(myEmployee.next()));

        }
```