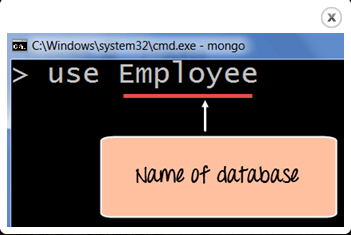
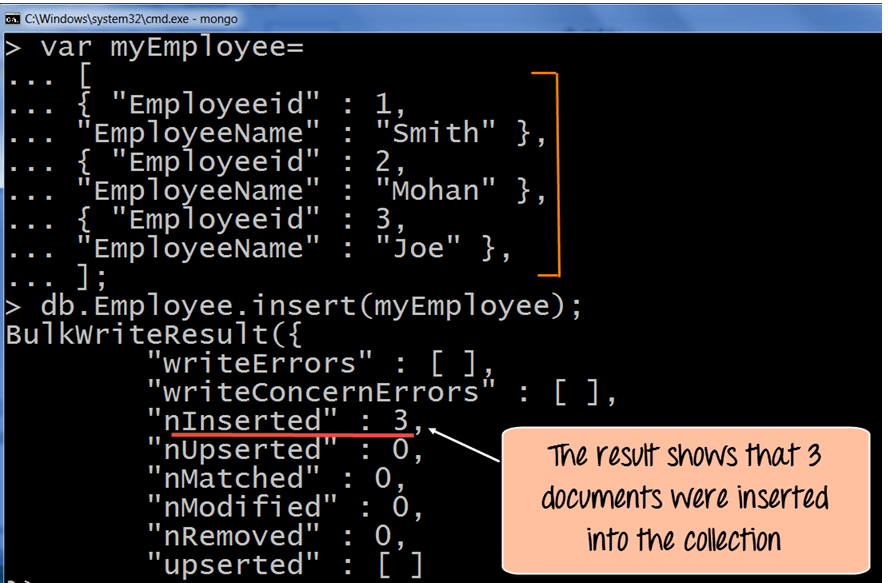
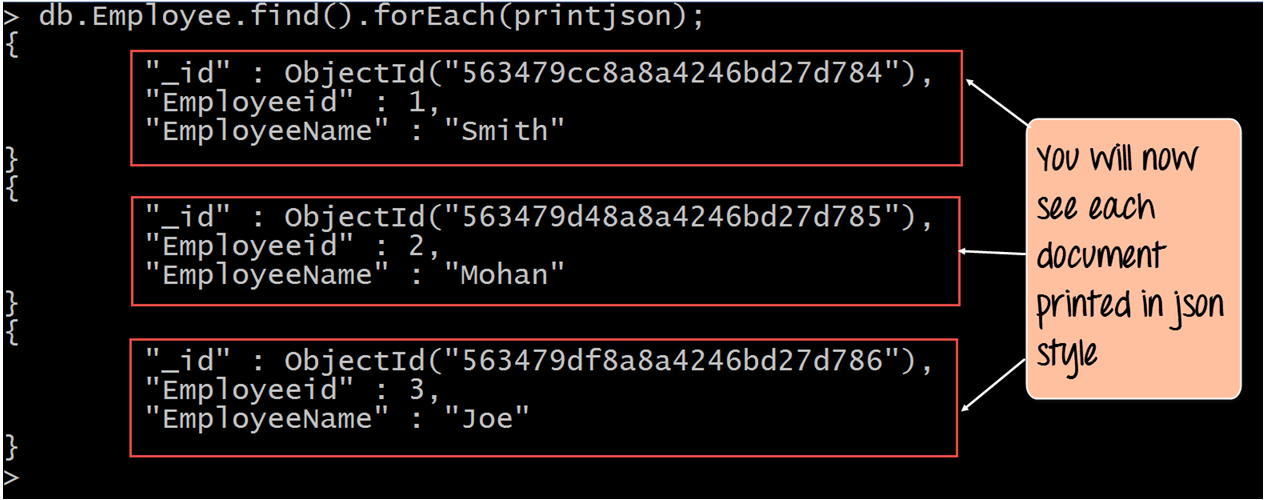
**MONGODB**

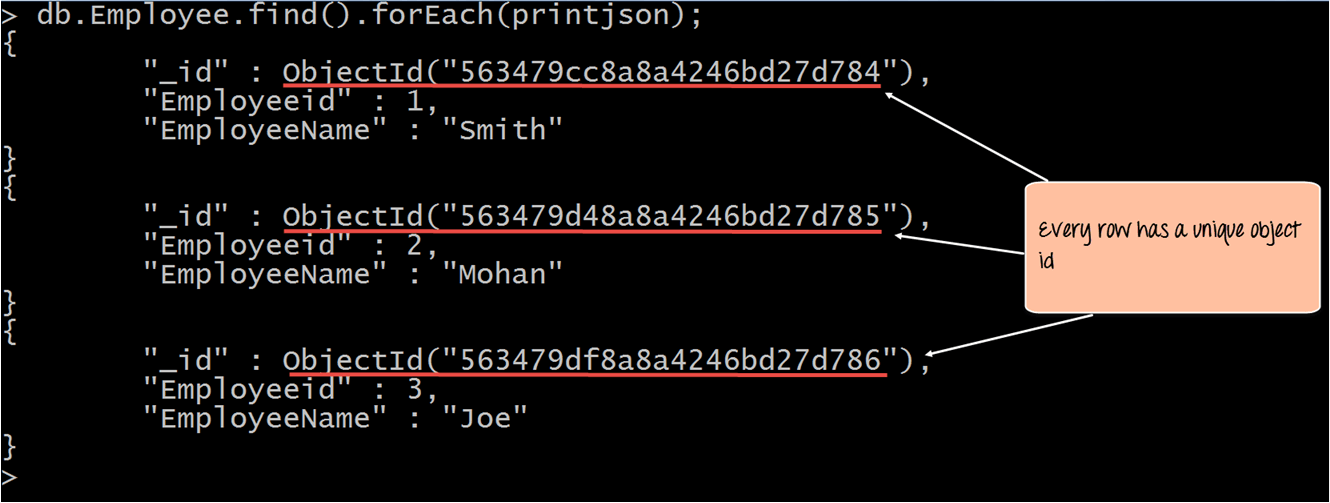
1. **What is MongoDB**
   1. MongoDB is a document database. Each database contains collections which in turn contains documents. Each document can be different with varying number of fields. The size and content of each document can be different from each other
   2. The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily
2. **Common terms in MongoDB**
   1. ***\_id:*** This is a field required in every MongoDB document. The \_id field represents a unique value in the MongoDB document. The \_id field is like the document's primary key. If you create a new document without an \_id field, MongoDB will automatically create the field.
   2. ***Collection:*** This is a grouping of MongoDB documents. A collection is the equivalent of a table which is created in any other RDMS such as Oracle or MS SQL. A collection exists within a single database. As seen from the introduction collections don't enforce any sort of structure
   3. ***Cursor:*** This is a pointer to the result set of a query. Clients can iterate through a cursor to retrieve results.
   4. **Database** : This is a container for collections like in RDMS wherein it is a container for tables. Each database gets its own set of files on the file system. A MongoDB server can store multiple databases.
   5. **Document** - A record in a MongoDB collection is basically called a document. The document in turn will consist of field name and values.
3. MongoDB Create & Insert Database
   1. Create database using “USE” Keyword
   2. Create database using insert() command
4. Add MongoDB Array using insert()

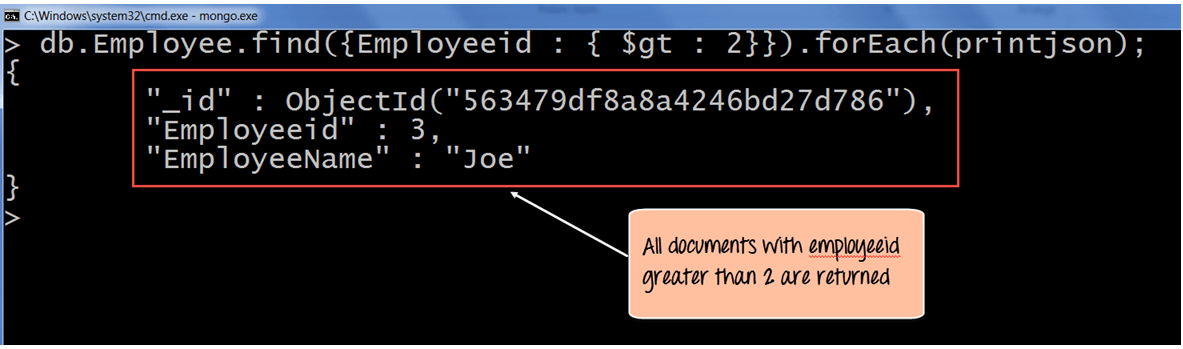


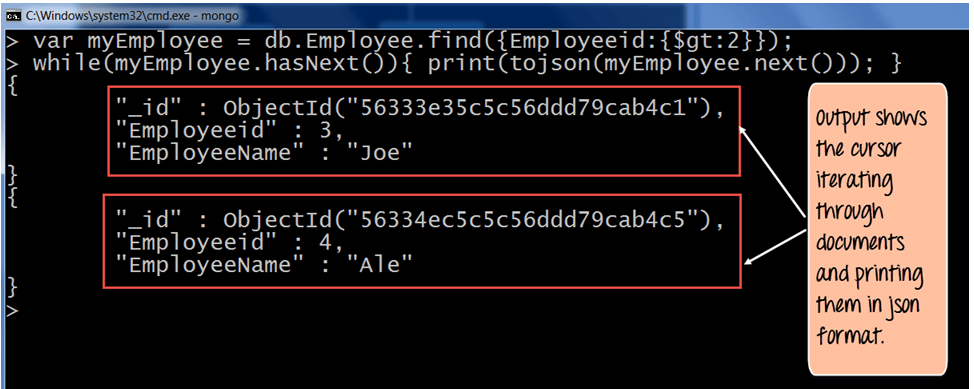
* 1. Printing in JSON format



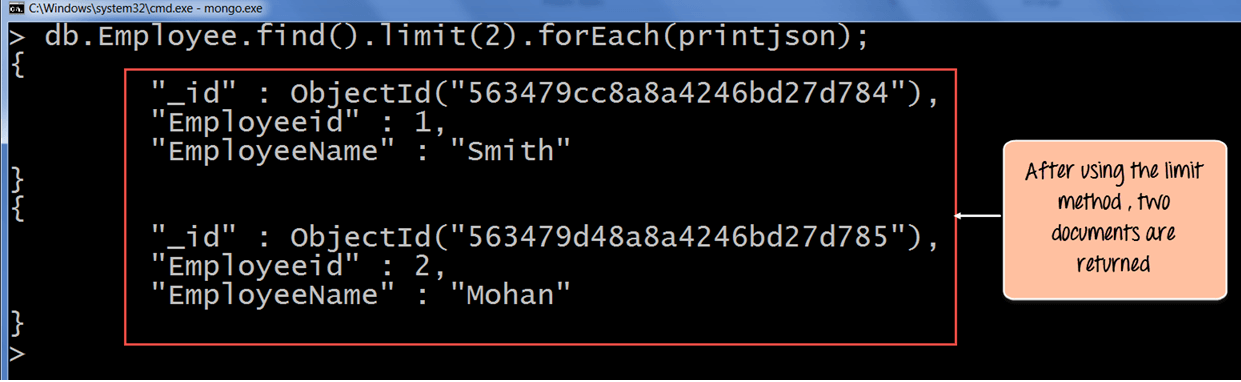
1. MongoDB Objectid()
   1. By default when inserting documents in the collection, if you don't add a field name with the \_id in the field name, then MongoDB will automatically add an Object id field.



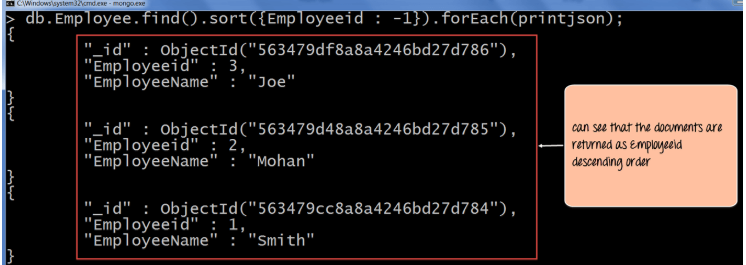
1. MongoDB Query Document using find()
2. MongoDB Cursor



1. MongoDB Query Modification using limit(), sort()
   1. Limit()
      1. This modifier is used to limit the number of documents which are returned in the result set for a query. The following example shows how this can be done.



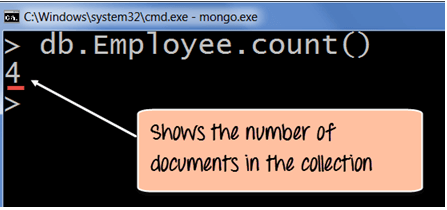
* 1. Orders

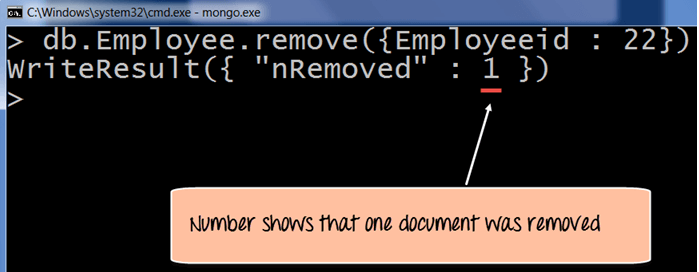


1: ascending

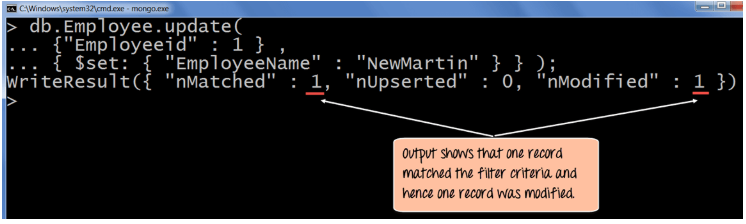
2: descending

1. MongoDB Count() & remove() function
   1. Count()

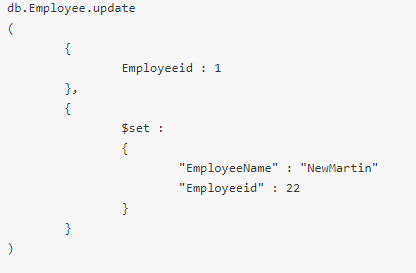


* 1. Remove()

1. Update() Document
   1. Upadate single value



* 1. Update Multiple value



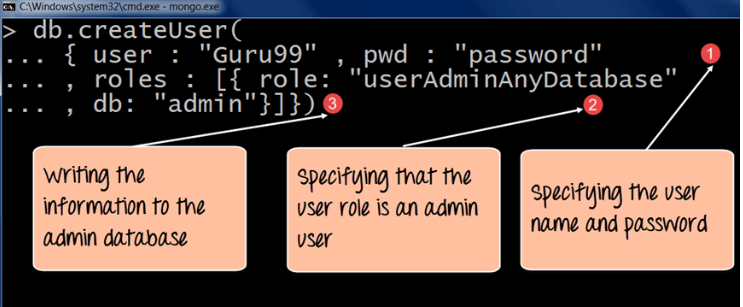
1. MongoDB Indexing, Monitoring & Backup
   1. MongoDb Sercurity
      1. MongoDB has the ability to define security mechanisms to databases. By default one wouldn't want everyone to have an open access to every database in MongoDB, hence the requirement for having some sort of security mechanism in MongoDB is important.
      2. Following are the best practices when implementing security in databases:
         1. Enable access control – Create users so that all applications and users are enforced to have some sort of authentication mechanism when accessing databases on MongoDB.
         2. Configure role based access control – Sometimes there can be a logical grouping of permissions which may be required, which can be clubbed in roles. Users can then be assigned to these roles.
         3. Try to configure MongoDB to use some sort of encryption protocol such as TLS or SSL. These protocols can be used to encrypt the traffic which flows between the client and the mongo DB environment
         4. Configure auditing – Administrators normally need to know who is doing what, which helps in analyzing problems later on. The best way is to enable auditing in MongoDB
         5. Run MongDB server instance with a separate user id which has access to the required resources on the server environment

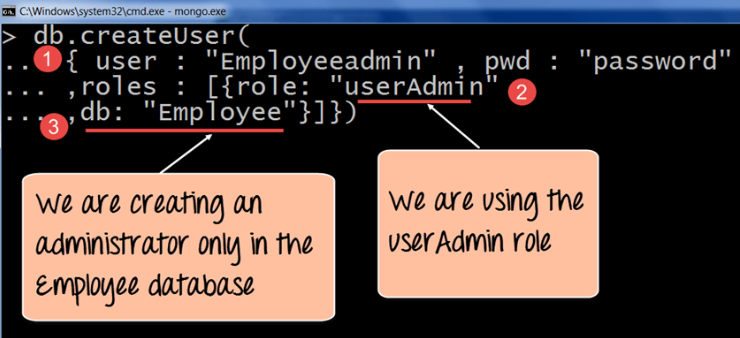
## MongoDB Indexing and Performance Considerations

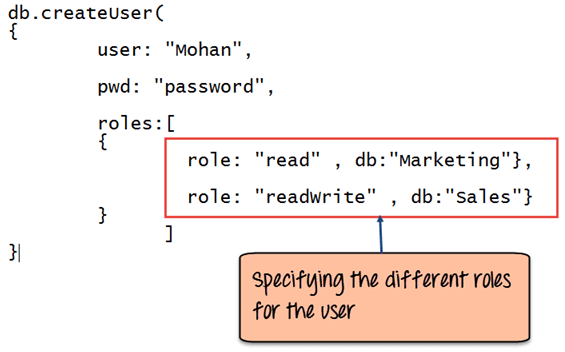
## Indexes are very important in any database and can be used to improve the efficiency of search queries in MongoDB. If you are continually performing searches in your document, then it's better to add indexes on the fields of the document that are used in the search criteria

* 1. MongoDB Backup Procedures
     1. When working with MongDB it is important to always ensure a backup procedure is in place in case the data within MongoDB gets corrupted for any reason
     2. Below are the backup mechanisms available from within MongoDB:
        1. **Backup by Copying Underlying Data Files** – This is probably the easiest mechanism , all that needs to be done is to copy the data files on which MongoDB resides and copy it to another location which ideally should be another server.
        2. **Backup a Database with mongodump** - The mongodump tool reads data from a MongoDB database and creates high fidelity BSON files. What needs to be taken into consideration is that if the data set is large in volume, then mongodump can be very resource intensive, so then to mitigate this problem, the utility should be run on a secondary server
        3. **MongoDB Cloud Manager Backup** - MongoDB Cloud Manager continually backs up MongoDB replica sets and sharded clusters by reading the oplog data from the MongoDB environment. MongoDB Cloud Manager can create a point in time recovery by storing oplog data so that it can create a restore at any moment in time for a particular replica set or sharded cluster
  2. MongoDB Monitoring
     1. Monitoring is one of the most critical administrative activities in MongoDB. This is because you can be more proactive by monitoring the environment for possible issues which could crop up
     2. Below are some of the ways of implementing monitoring:
        1. [mongostat](https://docs.mongodb.org/manual/reference/program/mongostat/) will tell you how many time database operations such as insert, query, update, delete, etc. actually occur on the server. This will give a good idea on how much the load the server is handling and will indicate whether you need additional resources on the server or maybe additional servers to distribute the load.
        2. [mongotop](https://docs.mongodb.org/manual/reference/program/mongotop/) tracks and reports the current read and write activity of a MongoDB instance, and reports these statistics on a per collection basis
        3. MongoDB provides a web interface that exposes diagnostic and monitoring information in a simple web page. One can browse to the below url on your local server to open the web administration utility [http://localhost:28017](http://localhost:28017/)

1. Create User in MongoDB & assign roles
   1. **Create an user has privileges over all the databases in MongoDB**

****

* 1. **Create an user has privileges on single database**
  2. **Managing Users**

****

1. **Mongodb authentication with Kerberos**
2. **MongoDB Replica Set** 
   1. Replication is referred to the process of ensuring that the same data is available on more than one Mongo DB Server. This is sometimes required for the purpose of increasing data availability.
   2. Another purpose of replication is the possibility of load balancing. If there are many users connecting to the system, instead of having everyone connect to one system, users can be connected to multiple servers so that there is an equal distribution of the load.
   3. In MongoDB, multiple MongDB Servers are grouped in sets called Replica sets. The Replica set will have a primary server which will accept all the write operation from clients. All other instances added to the set after this will be called the secondary instances which can be used primarily for all read operations.