27/5/20 MongoDB

MongoDB is a cross-platform document-oriented database program. classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server side Public license. MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need.

Key features of MongoDB

① Support ad hoc queries.
In mongodb you can search by field, range query and it also supports regular Expression searches.

② Indexing
you can index any field in a document

③ Replication
MongoDB ~~can perform~~ supports slave replication.
A master can perform reads and writes and a slave copies data from the master and can only be used for reads or backup (not writes).

④ Duplication of data
MongoDB can run over multiple servers.

The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.

⑤ <u>Load balancing</u>

It has an automatic load balancing configuration because of data placed in shards.

⑥ Supports map reduce and aggregation tools

⑦ uses javascript instead of procedures

⑧ It is a schema-less database written in C++.

⑨ provides high performance.

⑩ stores files of any size easily without complicating your stack.

⑪ Easy to administer in the case of failures.

⑫ It also supports:

* JSON data model with dynamic schemas
* Auto-sharing for horizontal scalability
* Built in replication for high availability
* Now a days many companies using mongo dB to create new types of application, improve performance and availability.

· Difference b/w mongodB structure and SQL structure.

SQL Terms/concepts	MongodB Terms/concepts
database	database
tables	collections
rows	documents (BSON)
columns	fields

Commands can be run on a mongo shell

db - returns the name of current database been used

use <dbname> - switches to specified db

db.help() - throws help commands

show dbs - Throws list of all database under mongodb

show users - Returns list of users

db.createCollection("") - used to create a collection

db.showcollection - show list of collections within a db.

db.collection.insert({<document>}) - used to insert a document to a collection db.collection.insertmany([<document>,<document>, ...]) - used to insert many documents to a single collection

db.collection.find({Key:Value, Key:Value}) - used to find specified documents from a collection

db.collection.update({<query>, <updated document>)

db.<collectionname>.remove() - To delete a collection from database.

use - mongodb command to switch to existing database or to create a database

29/5/20

creating connection to mongodb through Nodejs

Step1: download and install mongodb node module.

Eg: npm install mongodb

Step2: Include mongodb and create instance.

Eg: var mongodb = require("mongodb");

Step3: create instance of mongoclient of mongodb

Eg: Var mongoclient = mongodb. mongoclient;

Step4: create a url with mongodb protocol, server name and port number it is running

Eg: 'mongodb://localhost:27017/';

Step 5: create a connection to mongodb through mongoclient by passing the mongo db url.

Eg:
```
mogoclient.connect(url, function (err,
                        client) {
    // client object through which we get
connected to specified db
  })
```

Step 6: Through client object we could able to connect to specified database under mongo db.

Eg: `Var db = client.db ("onlineshopping");`

Step 7: Through db object we could get reference to required collection using db. collection () method.

Eg: `Var collection = db. collection ("userlist");`

Step 8: add the required command

```
Eg: collection.find().toArray(function
                       (err, result){
        ....
    })
```

3/6/20 **Session**

a session is time between login to logout. There are n number of node modules been supported through which we can implement sessions in any application

* Expression - session is one of the predifine module through which we can implement sessions within node Express server

following are the steps to be followed in order to implement sessions through "Express - session" in node module.

Step 1: download and install Expression - session module

```
npm install Express- session
```

Step 2: include Express - session module in app.js module

```
Var session = require("Express - session");
```

Step 3: Instantiate the session object by passing secret key, cookie status etc as an object

```
app.use (session({
            Keyboard cat
    secret : "<any secret key>",
    resave : false,
    saveUninitialized : true,
    cookie : {secure : true}
}));
```

Step 4: after the above steps, under each request object a session object will exist to which we can inject required data and validate.

4/6/20

Clustering

Clustering refers to a group of servers working together on one system to provide users with higher availability. using clusters we can reduce the dom time and outages by allowing another server to take over in ~~an~~ the event of an outage.

* clusters work like a group of ~~com~~ servers connected to a single system the moment a server experience a server outage, the work load is redistributed to server to another server before any downtime is Experienced.

⇒ Steps to get no. of CPU's of current system through OS module.

Step 1: create and install OS module

```
const os = require('os')
const cpuCount = os.cpus().length;
```

Implementing clusters in Nodejs

Step 1: download and install cluster module

Step 2: include the cluster module through require method

Step 3: cluster.fork() is a pre-defined method through which we can have a cluster copy of existing server.

Step 4: we can create a cluster copy only for the master cluster instance which can be checked through cluster.isMaster property.