**Mongo DB – 21st Oct 2019**

---------------------------------------------------------

It is unstructured, NoSQL database.

Stores the data in JSON format.

To store related info togetherly.

Unstructured Databases-

Documents (JSON) - **MongoDB**, SQL API(documentDB)

Key-Value/Tuple Pair - Azure table API, Redis Cash Database

Graph - Gremlin, Neo4j, (Adjest and vertises (Linked In))

Column family - Cassandra, Hadoop, Cloud data

It is open source driven by MongoDB community.

Onpremise NoSQL database that follows documents data storage.

Supports data indexing and partitioning (Sharding) for better performance.

Highly secured (multiple level authentication support)

It use BSON (Binary JSON) as the data storage format and JS as procedure and functions.

Installing MongoDB:

Install binary file .msi (installable), Run as a windows service.

Portable executable is available, download zip and extract. Need to run explicitly.

In the bin folder we can see mongodb server and client. (“MongoDB\Server\4.0\bin\”)

Server file: mongod.exe

Client Tool: mongo.exe (CUI)

GUI Client: VS code CosmosDB plugin, NoSQL booster for MongoDB, MongoCompass.

Migration Tools:

monogexport.exe

mongoimport.exe

mongodump.exe

mongorestore.exe

Installing BY ZIP:

Type below text in notepad:

"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe" --dbpath C:\Program Files\MongoDB\Server\4.0\data

Save this as

Mongodb://[hostname/ipaddress]:27017

mongodb://127.0.0.1:27017

Open cmd from mongo bin directory

And run

mongo.exe

use <dbname>

use eshop

* switched to db eshop done
* To check the db is created or not
  + db.stats();
* To get db collection names
  + db.getCollectionNames();

**Relational DB vs NoSQL:**

Database – database

Table – Collection/Container

Row – Document

Columns – Attributes

Every mongo dB document have a unique identifier with the name \_id.

A unique identifier

For Insert record:

insertOne()

insertMany()

**How to querying Data?**

We use find() for fetching data.

Conditions to fetch data:

Find(filter);

Find({id:3});

Find(filter, projection);

1. exclude
2. include

Sorting item

db.products.find({},{\_id:0, name:1, "availability": 1, "price": 1})

.sort({price:-1})

For ASC order use: 1

For DESC order use: -1

Query a fixed number of document

db.products.find({}, {\_id:0}).limit(4);

db.products.find().skip(4).limit(4);

Update documents/records:

db.colletion.update(filter, updateData, options)

db.products.update({availability:"Out of Stock"},

    {

        $set:{availability:"Available to Sale", quntity:10 }

    },

    { multi:true }

);

Add new attribute in existing record

db.products.update({quntity:10},

    {

        $set:{ expiryDate:"2019-09-21" }

    },

    { multi:true }

);

If update query not able to find this same record then insert new one.

db.products.update({name:"Pepsi"},

{

    $set:{id:12, quntity:4}

},

{upsert:true}

)

**How to use relational operators:**

Relational Operator: $le, $lte, $ge, $gte, $ne, $eq

Conditional Operator: $and, $or

Syntax:

(condition) AND (condition)

And(condition, condition)

Mongo $and(condition, condition)

**Array Operations:**

Iterate through an array of elements and push to array when use with $push

To avoid duplication we can use $addToSet

**db.students.update(**

**{name:"Arti"},**

**{$push:{marks:{$each:[25,30,35]}}}**

**)**

**Mongo DB Aggregation:**

Combine data from multiple documents.

3 types of aggregation:

Single purpose aggregation functions.

Count()

Distinct()

Aggregation pipeline.

An array of operation that executes one after another.

Result of one operation is the input of next operation.

We use operator like $match, $project, $group, $sort, $unwind etc.

Map-Reduce function (java script code)