**Mongo DB**

* Graphical user interface, text, application, email

  Description automatically generated
* In mongo DB the data is present in BSON format, which is same as JSON but is typed, meaning it has datatype associated to it like int, string, date, etc.
* Graphical user interface, text, application

  Description automatically generated
* The maximum size of one document in Mongo DB is 16MB.
* **How to start**
  + Open terminal and type mongod.exe to start the server.
  + Open another terminal and type mongo to start the shell.
* **Commands**
  + **use**
    - Syntax – **use** dbname
    - this command creates a new db if not present already by the given name and then switches to it.
    - if the db is already present then it switches to it.
  + **db**.collectionName.**insertMany()**/**db**.collectionName.**insertOne()**
    - insertMany() is used to add an array of objects in a collection and insertOne() is used to add an object in an array.
    - db is to point to the current database.
    - We must give a collection name in which data is going to be added.
    - It will automatically add an object ID in the inserted record.
    - Eg - db.tours.insertOne({name:"The Forest Hiker",price:297,rating:4.7}).
  + **db**.collectionName.**find()**
    - It is just like select\*from of SQL.
    - It is used to display the current collection data.
    - Eg – db.tours.find().
    - If we find and display some specific info just add a filter object in the find function.
    - Eg – db.tours.find({name : “The Forest Hiker”}).
    - If we want to show the documents in the range, we need to add a query in an object
    - Eg – price less that equal to 500 -> db.tours.find({ price: {$lte:500} }) **here $lte is less then equal too and we have to add $ before every query.**
    - For adding multiple queries with and operation , Eg price < 500 and rating >=4.8 -> db.tours.find({ price: {$lt:500}},{rating:{$gte:4.8}}), **where , is used for and operation.**
    - For adding multiple queries with or operation , Eg price < 500 or rating >=4.8 -> db.tours.find({$or: [{ price: {$lt:500}},{rating:{$gte:4.8}}]}), **where $or is used for or operation.**
    - For displaying only one field of the output query use -> db.tours.find({$or: [{ price: {$lt:500}},{rating:{$gte:4.8}}]}{name:1}), here it will only show the name of output query.
  + **show dbs**
    - used to show all the available dbs.
  + **show collections**
    - Used to show all the collections.
  + **quit()**
    - Used to exit mongo DB.
  + **db**.collectionName.**updateOne()**
    - this is used to update only one first found document in a collection, for all documents update we can use updateAll().
    - Eg - db.tours.updateOne({name: "The Snow Adventure"}, {$set: {price:597}}), here by name we are telling which document we need to select and then by $set we tell which field we want to update.
    - In other words, first object is used to find the document and second is what we want to update. Eg ->db.tours.updateMany({price:{$gte:500},rating:{$gte:4.8}},{$set: {premium:true}}).
  + **db**.collectionName.**deleteOne()/db**.collectionName.**deleteMany()**
* **Mongoose**
  + It is a driver used to connect the mongo db to nodejs application.
  + Install it by **npm i mongoose**.
  + Graphical user interface

    Description automatically generated with medium confidence
* **Indexes**
  + A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. ... An index is a copy of selected columns of data, from a table, that is designed to enable very efficient search.
  + Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. The index is a type of data structure. It is used to locate and access the data in a database table quickly.