Unstructured Data Management

CIS8045: Assignment 01 (MongoDB)

Database: sample_mflix/Collection: Media
Sample Data: Mongo_1_Documents_CreateDB

PART-01:

Query 1: Find the total number of “CDs
ANS:db.Media.count({"Type": “CD"})

Query 2: Find the books with a length between 200 and 300 pages, and return the top 5 longest ones
ANS:db.Media.find({'Page': { '$gt': 200, '$lt': 300}}).sort({"Page": -1}).limit(5)

Query 3: Find all the books published by either “Apress” or "O'Reilly Media”;
ANS: db.Media.find({$or: [{"Publisher": "Apress"}, {"Publisher": "O'Reilly Media"}]})

Query 4: Find all the CDs with 2 tracks or 3 tracks;
ANS: db.Media.find({$or :[{"Tracklist.Track" : "2"}, {"Tracklist.Track": "3"}]})

Query 5: Find all the books published by “Apress” in 2014. And display their “Title”, “Publisher”, and “Year”;
ANS: db.Media.find({$and: [{"Publisher": "Apress"}, {"Year": 2014}]}, { Title: 1, Publisher: 1, Year: 1 })

Query 6: Find all the CDs with the New Price not greater than $16;
ANS: db.Media.find({ "Price.New": { $lte: 16 } })

Query 7: Find all the books with either “David Hows” or “Peter Membrey” (or both of them) as the authors;
ANS: db.Media.find({$and :[{$or :[{"Author" : "David Hows"}, {"Author" : "Peter Membrey"}]}]})

Query 8: Find the CD with a Track 1 of Title “Don’t Stop ’Til You Get Enough”.
ANS: db.Media.find({ "Tracklist.Track": "1", "Tracklist.Title": "Don't Stop 'Til You Get Enough" })

Query 9: Update data as follows:
-For each book published by “Apress”, set a price of $100;
-Increase the price of the book “MongoDB Basics” by 20%;
-Insert your name into the author list of the book “MongoDB Basics
ANS: a) db.Media.updateMany(
{"Publisher": "Apress"},
{$set:{"Price.New": 100}
},{"projection": {"Publisher": 1, "Title": 1, "Price.New": 1, "_id": 0}, "returnNewDocument": true})
b) db.Media.updateMany({"Title":"MongoDB Basics"}, {$mul: {"Price.New": 1.2}});
c) db.Media.updateMany({ Title: "MongoDB Basics" },{ $set: { "Author" : "Bhavin Patel" } });

PART-02:

Database: Enron/Collection: Emails
Sample Data: Enron

Query 1: Develop a single aggregation query to find all senders and the total number of emails that each of them sent;
ANS: db.Emails.aggregate([{ $group: { _id: "$sender", total: { $sum: 1 } } }])

Query 2: Develop a single Aggregation query to complete the following tasks. (Note that you need to develop a single query rather than separate queries).
First, find all emails sent after 2000-10-20;
Next, among the emails sent after 2000-10-20, find the earliest email sent by each of the unique senders; Next, list each of the unique senders and his/her earliest email date. The list should be in a descending order of the earliest email dates.
ANS: db.Emails.aggregate([
{$match:{date:{$gte: "2000-10-20"}}},
{$group:{_id:"$sender","Earliest_Email":{$min:"$date"}}},
{$sort:{"Earliest_Email":-1}},
{$project:{ _id:1, sender:1, Earliest_Email:1 }}])