**MongoDB Queries**

**Inserting documents into collections:**

1. **Adding documents into service requests.**

db.service\_requests.insertOne({

"complaint\_severity": "m",

"service\_agent\_name": "Sandeshn",

"service\_id": "400000",

"date\_of\_complaint": "2022-01-02",

"complaint\_description": "Buffering",

"emailid": "OoececBAnkd@yahoo.com"

})

1. **Adding a user into Users collection.**

db.users.insertOne({

"emailid": "[new\_user@gmail.com](mailto:new_user@gmail.com)",

"phoneno": "322-370-1228",

"name": "Ratan Harry",

"dob": "2001-11-12",

"password": "qwerty12#",

"account\_create\_date": "2010-10-07",

"no\_of\_profiles": "0",

"preferred\_language": "English",

"profiles": [],

"billing\_date": "2022-04-03",

"status": "Active",

"payment\_method":"DEBIT/CREDIT"

});

**Queries for features:**

1. **New user registers into the application** :

db.users.insertOne({"emailid": "new\_user@gmail.com","phoneno": "322-370-1228","name": "Ratan Harry","dob": new Date("2001-11-12"),"password": "qwerty12#", "account\_create\_date": new Date("2010-10-07"),"no\_of\_profiles": 0,"preferred\_language": "English","profiles": [],"billing\_date": new Date("2022-04-03"), "status": "Active","payment\_method":"DEBIT/CREDIT"})

1. **User adds a profile into his account:**

db.users.updateOne({"emailid":"new\_user@gmail.com"},{ "$push": {"profiles": {"profile\_name": "Profile1","service\_id": 80012, "profile\_history": [],"profile\_my\_list": [],"complaints": []}}})

**Updating no\_of\_profiles count:**

db.users.updateOne({ "emailid": "new\_user@gmail.com" },{ "$inc": { "no\_of\_profiles": 1 } })

1. **Deleting a Profile:**

db.users.update({"emailid":"[0omsUvRjUHZlzEPT@msn.com](mailto:0omsUvRjUHZlzEPT@msn.com)"},{$pull:{"profiles":{"profile\_name":"sush"}}})

**Updating the no\_of\_profiles:**

db.users.updateOne({ "emailid": "0omsUvRjUHZlzEPT@msn.com" },{ "$inc": { "no\_of\_profiles": -1 } })

1. **Adding a movie to Profile watching list:**

var movie = db.movies.findOne({"Title":"Guardians of the Galaxy"})

var movieId = movie.\_id

var movie\_rank = movie.Rank

db.users.updateOne({ "profiles.profile\_name" : "Profile1", "emailid":"new\_user@gmail.com"}, { "$push": { "profiles.$.profile\_my\_list": { "imdb\_rank": movie\_rank, "movie\_id":movieId}}})

1. **Adding a movie to Profiles’ history:**

var movie = db.movies.findOne({"Title":"Prometheus"})

var movieId = movie.\_id

var movie\_rank = movie.Rank

db.users.updateOne({ "profiles.profile\_name" : "Profile1", "emailid":"new\_user@gmail.com"}, { "$push": { "profiles.$.profile\_history": { "view\_timestamp":new Date(Date.now()), "imdb\_rank": movie\_rank, "movie\_id":movieId}}})

1. **Deleting a movie from Profiles’ watching list:**

var movie = db.movies.findOne({"Title":"Guardians of the Galaxy"})

var movieId = movie.\_id

var movie\_rank = movie.Rank

db.users.update({"emailid":"new\_user@gmail.com", "profiles.profile\_name": "Profile1"}, { $pull: {"profiles.$.profile\_my\_list": {imdb\_rank: movie\_rank} } }, { multi: true })

1. **Deleting a movie from Profiles’ history:**

* Delete entire history

db.users.updateOne({"emailid":"new\_user@gmail.com","profiles.profile\_name":"Profile1"},{$unset:{"profiles.$.profile\_history":""}},{multi:true})

* Delete a specific movie from the history

var movie = db.movies.findOne({"Title":"Guardians of the Galaxy"})

var movie\_rank = movie.Rank

db.users.update({"emailid":"new\_user@gmail.com", "profiles.profile\_name": "Profile1"}, { $pull: {"profiles.$.profile\_history": {imdb\_rank: movie\_rank} } }, { multi: true })

1. **Adding a new movie:**

db.movies.insertOne({"imdb\_rank": "1001", "Title": "Fantastic Beasts - The Secrets of Dumbledore", "Genre": ["Action", "Adventure", "Sci-Fi"], "Description": "Professor Albus Dumbledore knows the powerful, dark wizard Gellert Grindelwald is moving to seize control of the wizarding world. ", "Director": "David Yates", "Actors": "", "Year": "2022", "Runtime (Minutes)": "121", "Rating": "8.1", "Votes": "757074", "Revenue (Millions)": "333.13", "Metascore": "80"})

1. **Deleting a movie from Movie Collection:**

db.movies.deleteOne({"Title":"Fantastic Beasts - The Secrets of Dumbledore"});

1. **Creating a Service request:**

db.service\_requests.insertOne({service\_id:400019, date\_of\_complaint:new Date(Date.now()), complaint\_description:"Problems with the subtitles or captions, including quality or availability", emailid:"[uqCdy@hotmail.com](mailto:uqCdy@hotmail.com)"})

**Service agent acknowledges the request raised:**

db.service\_requests.updateOne({service\_id:400019}, {$set:{complaint\_severity: "h", service\_agent\_name:"Preetha"}},{upsert:true})

**Adding a reference to this complaint raised in the data of the profile**

var serviceid = db.service\_requests.findOne({“service\_id”:400019})

var complaintId = serviceid.\_id

db.users.updateOne({ "profiles.service\_id" : 400019, "emailid":"uqCdy@hotmail.com"}, { "$push": { "profiles.$.complaints”: {\_id :complaintId}}});

1. **Deleting a user:**

db.users.deleteOne( { emailid : "2Q@msn.com" } )

1. **Displaying a user bill:**

db.users.find({emailid:'[3Gx@aol.com](mailto:3Gx@aol.com)'},{billing\_date:1,status:'active',payment\_method:1,emailid:1,\_id:0});

**Aggregations and Find Queries:**

1. **To find directors who have directed more than 1 movie**

db.movies.aggregate([{ "$group":{ "\_id" : "$Director","count" : {"$sum" : 1}}}, {"$match" : { "count" : { "$gte" : 2}}}])

1. **To find which movies that were released between the years 2018 to 2020**

db.movies.find({"Year": {"$gte": "2016", "$lte": "2018"}},{Title:1, Description:1, Year:1,\_id:0}).pretty()

1. **Showing Deactivated Users**

db.users.find({"status":"Deactivated"},{ emailid : 1,name:1, status:1})

1. **Actors who have performed in more than 1 movie.**

db.movies.aggregate([{ "$unwind" : "$Actors"},{ "$group":{ "\_id" : "$Actors","count" : {"$sum" : 1}}}, {"$match" : { "count" : { "$gt" : 1}}}])

1. **Getting the user details who raised the service request**

db.service\_requests.aggregate([{"$lookup" : {"from" : "users", "localField" : "emailid", "foreignField" : "emailid","as" : "users"}}, {$project : {"users.password":0, "users.dob":0, "users.billing\_date":0, "users.status":0, "users.profiles":0, "users.payment\_method":0}}])

1. **Top 5 rated movies**

db.movie.aggregate([{$sort:{Rating:-1}},{$limit:5},{$project:{"imdb\_rank":1, "Title":1,"Rating":1,"Votes":1}}])

1. **Types of service complaints raised by the user with their counts:**

db.service\_requests.aggregate([{ "$group":{ "\_id" : "$complaint\_description","count" : {"$sum" : 1}}}])

1. **Generate the next bill due date:**

db.users.aggregate([{$project:{"next\_bill\_due":{$dateAdd:{startDate:"$billing\_date", unit:"month", amount: 1}},name:1,billing\_date:1,amount:{$literal:"$19.95"}}},{$sort:{billing\_date:-1}},{$limit:5}])

1. **Daily service request counts :**

db.service\_requests.aggregate([{ "$group":{ "\_id" : "$date\_of\_complaint","count" : {"$sum" : 1}}}])

1. **Movie language preferred by the user with count**

db.users.aggregate([{"$group":{"\_id":"$preferred\_language","count":{"$sum":1}}}])

1. **View profiles my list**

db.users.aggregate([{$project:{"profiles":1,"\_id":0}},{$unwind:"$profiles"},{$unwind:"$profiles.profile\_history"}, {$lookup:{from:"movies", localField:"profiles.profile\_history.movie\_id",foreignField:"\_id", as:"movie\_title"}}, {$project:{"profiles.profile\_name":1,"movie\_title.Title":1,"\_id":0}}])