

Project Mongodb

Msc Data Analytics



May 13, 2019

reenu rathi

A00258770

Table of Contents

[**Introduction: 2**](#_Toc8594177)

[**Description of project: 3**](#_Toc8594178)

[**Queries with Arrays and Operators: 3**](#_Toc8594179)

[**Queries with Aggregation Framework: 10**](#_Toc8594180)

[**Queries of CRUD: 18**](#_Toc8594181)

[**You Tube Screencast links: 25**](#_Toc8594182)

[**References: 25**](#_Toc8594183)

# 

# Introduction:

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

**Database**

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

**Collection**

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

**Document**

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

The following table shows the relationship of RDBMS terminology with MongoDB.

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB** |
| Database | Database |
| Table | Collection |
| Tuple/Row | Document |
| column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (Default key \_id provided by mongodb itself) |
| **Database Server and Client** | |
| Mysqld/Oracle | mongod |
| mysql/sqlplus | mongo |

# Description of project:

The main aim of the proejct to demonstrate the MongoDb syntax and understand the difference between a relational database and noSql database working. We have provided the topic and table with the column details to prepare the database for further manupulation

This document will attempt to cover these following feature of MongoDB

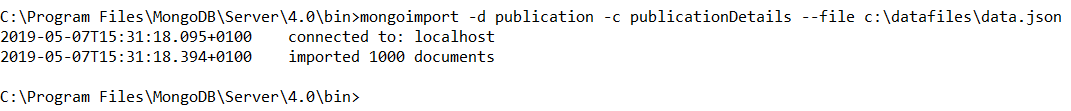
* MongoDB CRUD(Create, Retrieve, Update, Delete)
* MongoDB Aggregatiion pipeline
* MongoDB Array ,Embedded doucments and Operators.

**Publication** (\_Id, Name, Address, County, Web\_Site, Frequency (*Daily,Weekly,Monthly*), Readership(i.e. *quantity normally sold*), Circulation (*Local, Provincial, National*), Selling\_Price, **Format** (Newspaper, Television, Broadsheet, Tabloid, Online), **Audience** (Child,Youth, Adult, Retired, Others), **Specialism** (Politics, Sport, Science, Technology, Arts), **Revenue** (breakdown of revenue from different sources for the publication such as Hardcopy Sales, Online Sales – embedded document with **Sale**Hardcopy, **sale**Online

**Importing the json data file:**

We created the database named publication and collection named publicationDetails to preform furthers tasks.

mongoimport -d publication -c publicationDetails --file c:\datafiles\data.json



# Queries with Arrays and Operators:

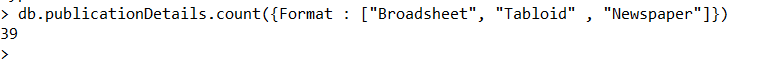
**Question 1: List the Id, names, address and format of publishers whose format of publication is exactly the following three Broadsheet, Tabloid and Newspaper in the sequence.**

db.publicationDetails.find({Format : ["Broadsheet", "Tabloid", "Newspaper"]},{Name : 1, Address : 1, Format : 1})



**Question 2: How many publishers have format details that is exactly in the sequence of Broadsheet, Tabloid and Newspaper.**

db.publicationDetails.count({Format : ["Broadsheet", "Tabloid" , "Newspaper"]})



**Question 3: List the Id, names, address and format of publishers whose format of publication is exactly Tabloid as a format.**

db.publicationDetails.find({Format : ["Tabloid"]},{Name : 1, Address : 1, Format : 1})



**//Count how many documents are retrieving//**

db.publicationDetails.count({Format : ["Tabloid"]},{Name : 1, Address : 1, Format : 1})



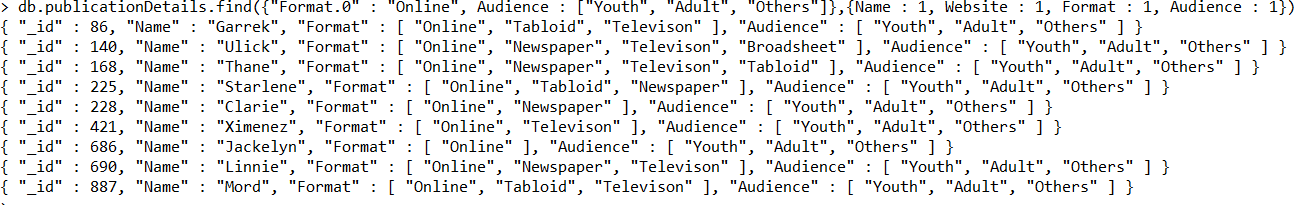
**Question 4: List the Id, Name, Website, Format and audience for the publisher who has Tabloid format and whose audience is only young people.**

db.publicationDetails.find({Format : ["Tabloid"], Audience : ["Youth"]},{Name : 1, Website : 1, Format : 1, Audience : 1})



**Question 5**: **List the Id, Name, Website, Format and audience for the publisher who has Online as first of its format and whose audience is exactly as follows Young, Adult and Others.**

db.publicationDetails.find({"Format.0" : "Online", Audience : ["Youth", "Adult", "Others"]},{Name : 1, Website : 1, Format : 1, Audience : 1})



**Question 6: List the Id, Name, Website, Audience and specialism for the publisher who has Child as first of its audience and whose specialism includes Art.**

db.publicationDetails.find({"Audience.0" : "Child", Specialism : "Art"},{Name : 1, Website : 1, Audience : 1, Specialism : 1})



**//Count how many documents are retrieving//**

db.publicationDetails.count({"Audience.0" : "Child", Specialism : "Art"},{Name : 1, Website : 1, Audience : 1, Specialism : 1})



**Question 7: List the Id, Name, Website, Audience and specialism for the publisher who has Child as first of its audience and Art first of its specialism.**

db.publicationDetails.find({"Audience.0" : "Child", "Specialism.0" : "Art"},{Name : 1, Website : 1, Audience : 1, Specialism : 1})

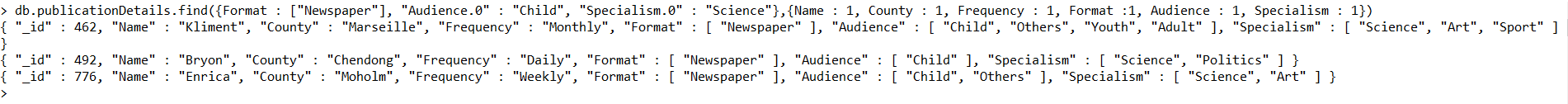


db.publicationDetails.count({"Audience.0" : "Child", "Specialism.0" : "Art"},{Name : 1, Website : 1, Audience : 1, Specialism : 1})



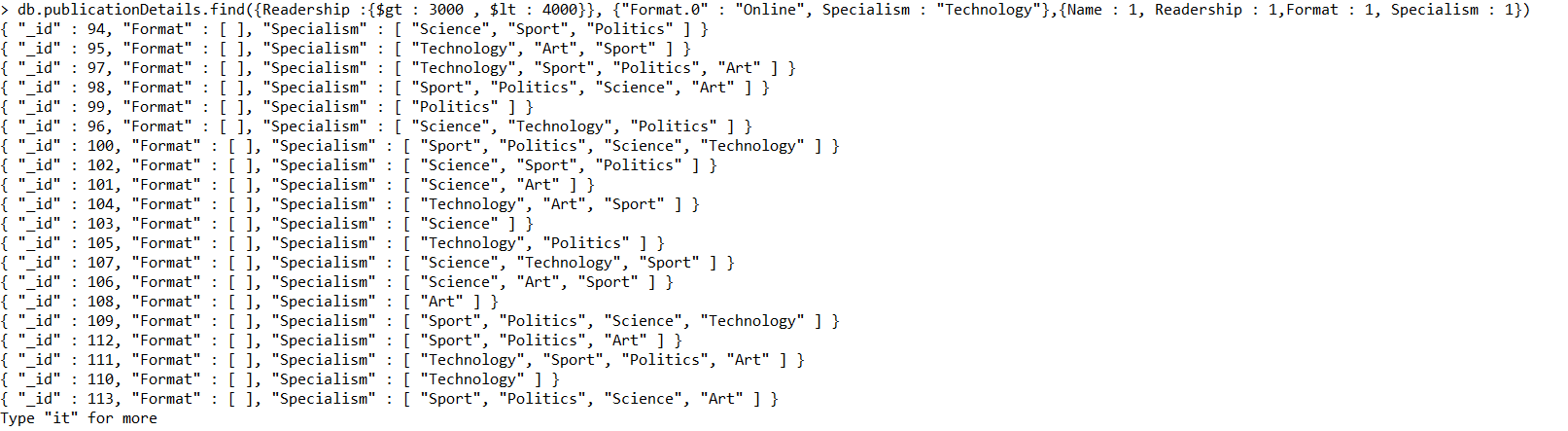
**Question 7: List the Id, Name, County, Frequency, Audience, Format and specialism for the publisher who has Child as first of its audience and Science first of its specialism and format as Newspaper only.**

db.publicationDetails.find({Format : ["Newspaper"], "Audience.0" : "Child", "Specialism.0" : "Science"},{Name : 1, County : 1, Frequency : 1, Format :1, Audience : 1, Specialism : 1})



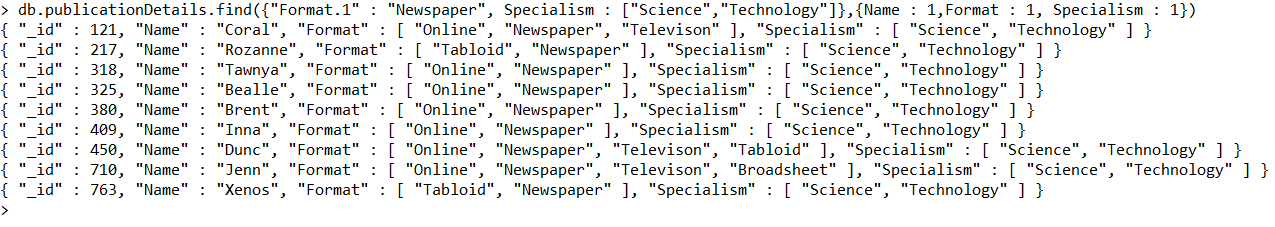
**Question 8: List the id and name readership, format and specialism of publishers whose and readership between 3000 to 4000 and Online as of its first format of publishing and includes Technology as specialism**

db.publicationDetails.find({Readership :{$gt : 3000 , $lt : 4000}}, {"Format.0" : "Online", Specialism : "Technology"},{Name : 1, Readership : 1,Format : 1, Specialism : 1})



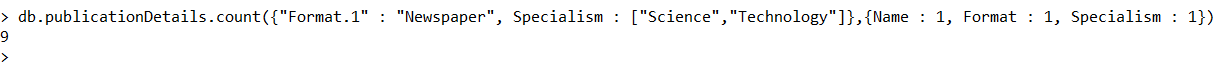
**Question 9: List the id and name, format and specialism of publishers whose has Newspaper as of its second format of publishing and includes Science and Technology as specialism in exactly in same sequence.**

db.publicationDetails.find({"Format.1" : "Newspaper", Specialism : ["Science","Technology"]},{Name : 1, Format : 1, Specialism : 1})



**//Count how many documents are in the list//**

db.publicationDetails.count({"Format.1" : "Newspaper", Specialism : ["Science","Technology"]},{Name : 1, Format : 1, Specialism : 1})



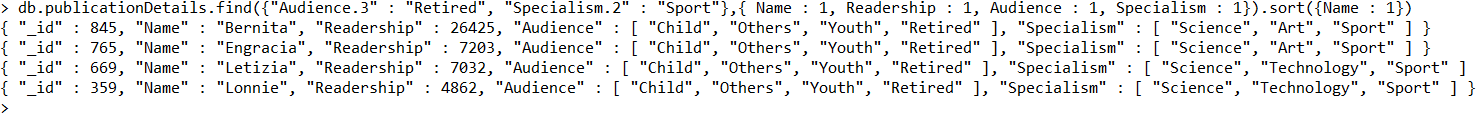
**Question 10: List the id, name, audience and specialism of publishers whose has adult as its second audience and includes sport as its in third specialism. Sort the result by name in descending order.**

db.publicationDetails.find({"Audience.1" : "Adult", "Specialism.2" : "Sport"},{\_Id : 1, Name : 1, Audience : 1, Specialism : 1}).sort({Name : -1})



**Question 11: List the id, name, readership, audience and specialism of publishers whose has Retired as its forth audience and includes sport as its third specialism. Sort the result by name in ascending order.**

db.publicationDetails.find({"Audience.3" : "Retired", "Specialism.2" : "Sport"},{ Name : 1, Readership : 1, Audience : 1, Specialism : 1}).sort({Name : 1})



**Question 12: List the id, county, website and audience details of publisher who has child and adult as audience among their list of audience value – sequence does not matter. Sort the result by county in descending order.**

db.publicationDetails.find({Audience : { $all : ["Adult", "Child"]}}, { County : 1, Audience : 1, Website : 1}).sort({County : -1})



**Question 13: List the id, county, website and specialism details of publisher who has art and sport among their list of specialism value – sequence does not matter. Sort the result by county in ascending order.**

db.publicationDetails.find({Specialism : { $all : ["Art", "Sport"]}}, { County : 1, Specialism : 1, Website : 1}).sort({County : 1})



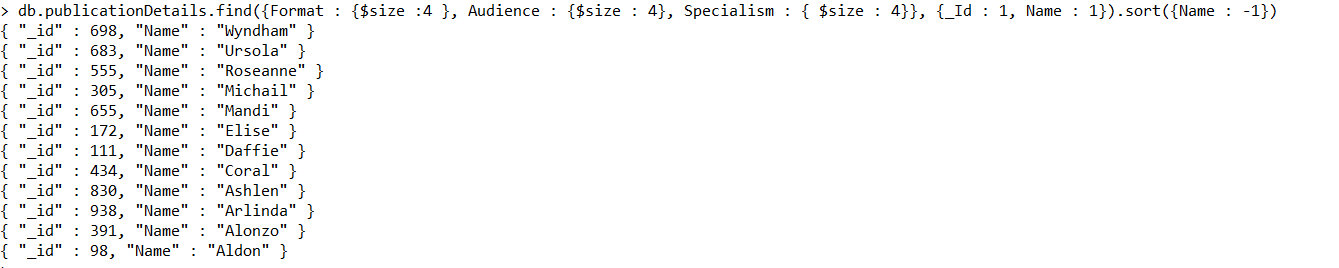
**Question 14: List the id, name, county, website and specialism details of publisher whose specialism value has 3 values in it. Sort the result by name in ascending order.**

db.publicationDetails.find({Specialism : { $size : 3}}, { Name : 1, County : 1, Specialism : 1, Website : 1}).sort({Name : 1})



**Question 15: List the id and name of the publisher whose has exactly four values in format, audience and specialism. Sort the result by name in descending order.**

db.publicationDetails.find({Format : {$size :4 }, Audience : {$size : 4}, Specialism : { $size : 4}}, {\_Id : 1, Name : 1}).sort({Name : -1})



**Question 16: How many publishers has exactly two formats, three audience and four specialisms.**

db.publicationDetails.count({Format : {$size : 2}, Audience : {$size : 3}, Specialism : { $size : 4}})



**Question 17: List second top document who has readership greater than 6000 from publisher collection.**

db.publicationDetails.find({Readership : {$gt:6000}}).limit(1).skip(1).pretty()



# Queries with Aggregation Framework:

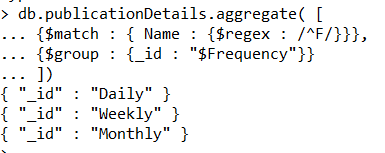
**Question 18: list the different frequency of publication for the publisher whose name starts with letter "F".**

db.publicationDetails.aggregate( [

{$match : { Name : {$regex : /^F/}}},

{$group : {\_id : "$Frequency"}}

])



**Question 19: For Each publisher whosae name includes "eo", display the name of publisher and the count of same names.**

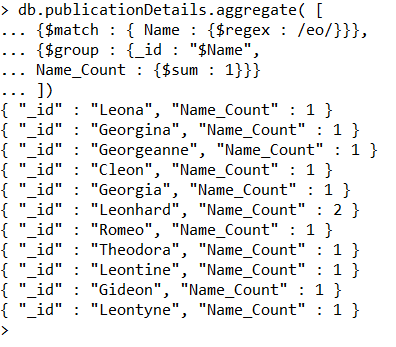
db.publicationDetails.aggregate( [

{$match : { Name : {$regex : /eo/}}},

{$group : {\_id : "$Name",

Name\_Count : {$sum : 1}}}

])



**Question :20 find most popular audience type in terms of revenue thorugh online sale by selling price which varies from 5000 to 20000. Only consider publisher who has readership of 4500 or more.**

db.publicationDetails.aggregate([

{$match : {"Revenue.saleOnline" : {$gt : 0}, Selling\_Price : {$gte : 5000, $lte : 20000}, Readership : {$gte : 4500}}},

{$unwind : "$Audience"},

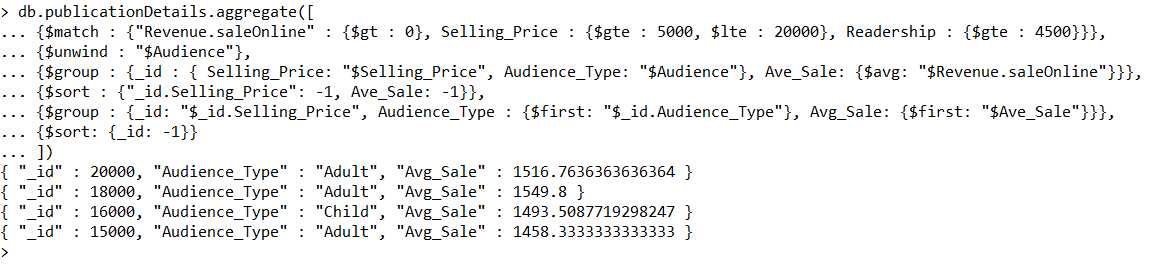
{$group : {\_id : { Selling\_Price: "$Selling\_Price", Audience\_Type: "$Audience"}, Ave\_Sale: {$avg: "$Revenue.saleOnline"}}},

{$sort : {"\_id.Selling\_Price": -1, Ave\_Sale: -1}},

{$group : {\_id: "$\_id.Selling\_Price", Audience\_Type : {$first: "$\_id.Audience\_Type"}, Avg\_Sale: {$first: "$Ave\_Sale"}}},

{$sort: {\_id: -1}}

])



**Question :21 For each format and within audience catogory, list the name , audience, format and maximum readership for that format and audience type combination.? Only display results for which the maximum readership exceeds 99k . sort by the maximum readership in ascending order. Considering that pooint aggregation pipeline only process the relevent data.**

db.publicationDetails.aggregate([

{$unwind : "$Format"},

{$unwind : "$Audience"},

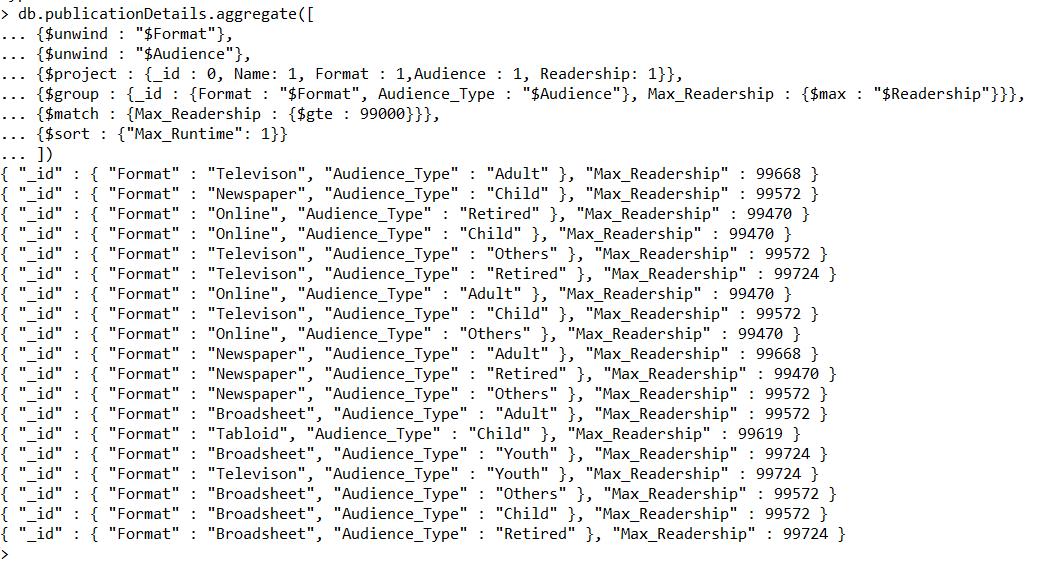
{$project : {\_id : 0, Name: 1, Format : 1,Audience : 1, Readership: 1}},

{$group : {\_id : {Format : "$Format", Audience\_Type : "$Audience"}, Max\_Readership : {$max : "$Readership"}}},

{$match : {Max\_Readership : {$gte : 99000}}},

{$sort : {"Max\_Runtime": 1}}

])



**Question 22: find most popular specialism type in terms of revenue thorugh hardcopy sale by selling price which varies from 15000 to 20000. Only consider publisher who has readership of 4500 or more.**

db.publicationDetails.aggregate([

{$match : {"Revenue.saleHardcopy" : {$gt : 0}, Selling\_Price : {$gte : 15000, $lte : 20000}, Readership : {$gte : 4500}}},

{$unwind : "$Specialism"},

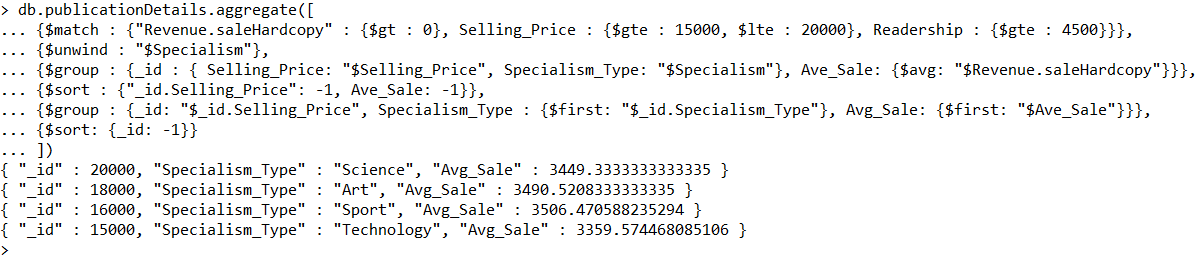
{$group : {\_id : { Selling\_Price: "$Selling\_Price", Specialism\_Type: "$Specialism"}, Ave\_Sale: {$avg: "$Revenue.saleHardcopy"}}},

{$sort : {"\_id.Selling\_Price": -1, Ave\_Sale: -1}},

{$group : {\_id: "$\_id.Selling\_Price", Specialism\_Type : {$first: "$\_id.Specialism\_Type"}, Avg\_Sale: {$first: "$Ave\_Sale"}}},

{$sort: {\_id: -1}}

])



**Question: 23 For each circulation type, indicate the number of county, the average selling price and lowest readership of publishers?Display only documents in the results that have an average selling price more or 18000.**

db.publicationDetails.aggregate([

{$group : {\_id : "$Circulation",

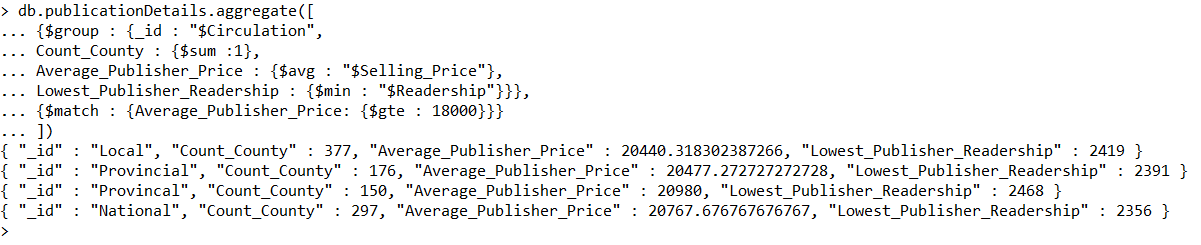
Count\_County : {$sum :1},

Average\_Publisher\_Price : {$avg : "$Selling\_Price"},

Lowest\_Publisher\_Readership : {$min : "$Readership"}}},

{$match : {Average\_Publisher\_Price: {$gte : 18000}}}

])



**Question :24 For each selling price value between 18000 to 20000, indicate the number of publishers, the average revenue by online sales and min revenue by hardcopy? Output the results to collection called Publisher\_Calc\_By\_Price. Show all results in the Publisher\_Calc\_By\_Price collection.**

db.publicationDetails.aggregate([

{$match : {"Selling\_Price": {$gte : 18000, $lte : 20000 }}},

{$group : {\_id : "$Selling\_Price",

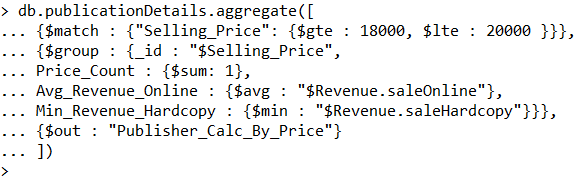
Price\_Count : {$sum: 1},

Avg\_Revenue\_Online : {$avg : "$Revenue.saleOnline"},

Min\_Revenue\_Hardcopy : {$min : "$Revenue.saleHardcopy"}}},

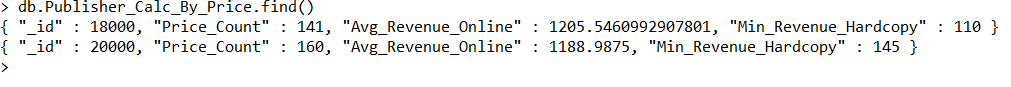
{$out : "Publisher\_Calc\_By\_Price"}

])



**//Confirm the aggregation has been succesful//**

db.Publisher\_Calc\_By\_Price.find()



**Question : 25 For those specialism who has publisher count more than 30, list the publisher specialism and publisher count. Sort the the result in ascending order by publisher count.**

db.publicationDetails.aggregate([

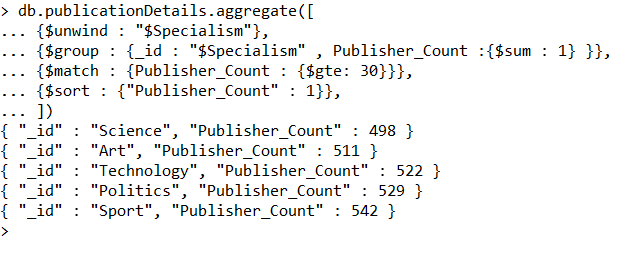
{$unwind : "$Specialism"},

{$group : {\_id : "$Specialism" , Publisher\_Count :{$sum : 1} }},

{$match : {Publisher\_Count : {$gte: 30}}},

{$sort : {"Publisher\_Count" : 1}},

])



**Question :26 Assume daliy publisher stats for a publisher can be calculated by adding the revenue through online and hardcopy and readership. Consider selling price in 15500 and 15000 that have exactly 4 formats and display the name sale through online and hardcopy and readership and daily publisher stats.**

db.publicationDetails.aggregate([

{$match: {Selling\_Price : {"$in" : [15500, 16000]}, Format : {$size:4}}},

{$project : {\_id : 0, Name : 1, "Revenue.saleOnline" : 1, Readership : 1, "Revenue.saleHardcopy" : 1,

Daily\_Publisher\_Stats : {$add : ["$Revenue.saleOnline", "$Readership", "$Revenue.saleHardcopy"]}}}

])



**Question :27 Assume that daliy stats for a publisher on (i) the revenue stats are equal to the revenue achiveing by online sales and (ii) readership stats are just readership count recorded. Consider frequency not in weekly and monthly and audience size is exactly 4 display the name online sale revenue, readership ,revenue stats and readership stats.**

db.publicationDetails.aggregate([

{$match : { Frequency : {"$nin" : ["Weekly", "Monthly"]}, Audience : {$size : 4}}},

{$project : {\_id:0, Name:1, "Revenue.saleOnline": 1, Readership : 1,

Revenue\_Stats : "$Revenue.saleOnline",

Readership\_Stats : "$Readership"}}

])



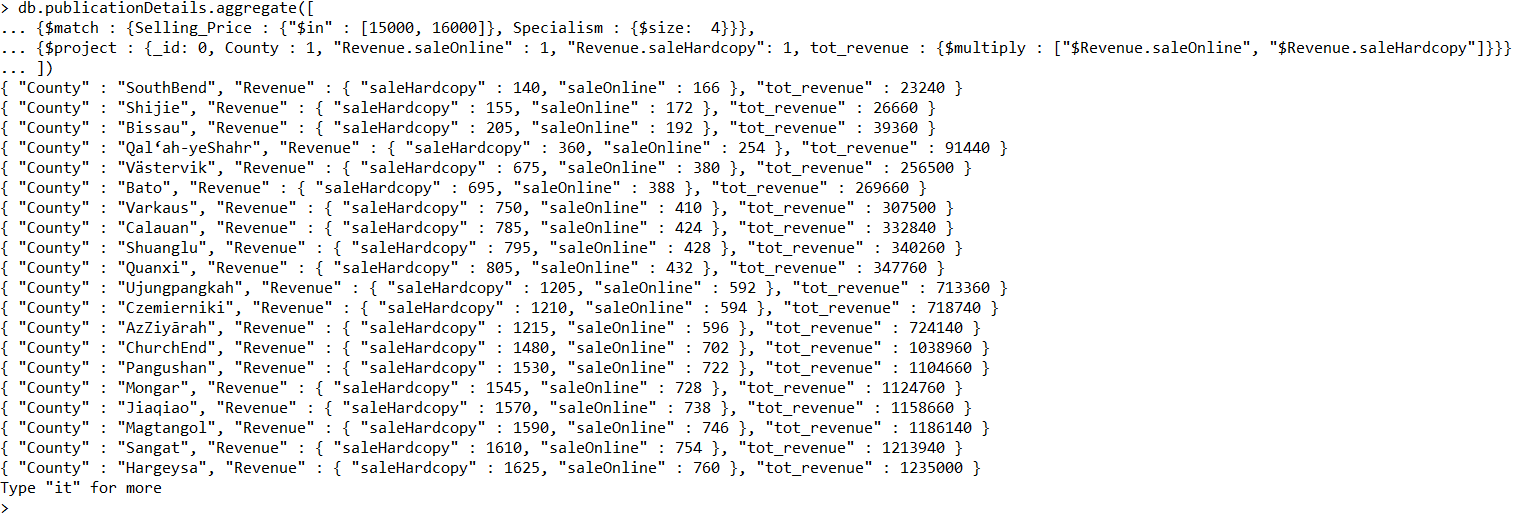
**Question :28 Assume that total earning for a publisher can be calcuated by multipling the revenue by online sale and hardcopy. Consider publisher whose selling price in 15000 and 16000 and has exactly 4 specialism , display the county earning ny onloine and hardcopy and total earning of the publisher.**

db.publicationDetails.aggregate([

{$match : {Selling\_Price : {"$in" : [15000, 16000]}, Specialism : {$size: 4}}},

{$project : {\_id: 0, County : 1, "Revenue.saleOnline" : 1, "Revenue.saleHardcopy": 1, tot\_revenue : {$multiply : ["$Revenue.saleOnline", "$Revenue.saleHardcopy"]}}}

])



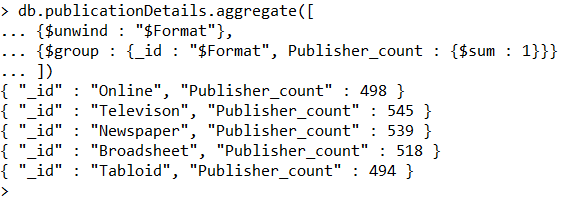
**Question :29 How many publisher fall under each of the format category mentioned in the publisher collection.**

db.publicationDetails.aggregate([

{$unwind : "$Format"},

{$group : {\_id : "$Format", Publisher\_count : {$sum : 1}}}

])



**Question :30 For each frequency type, indicate the number of county, the average selling price and maximum readership of publishers?Display only documents in the results that have an average selling price more or 18000.**

db.publicationDetails.aggregate([

{$group : {\_id : "$Frequency",

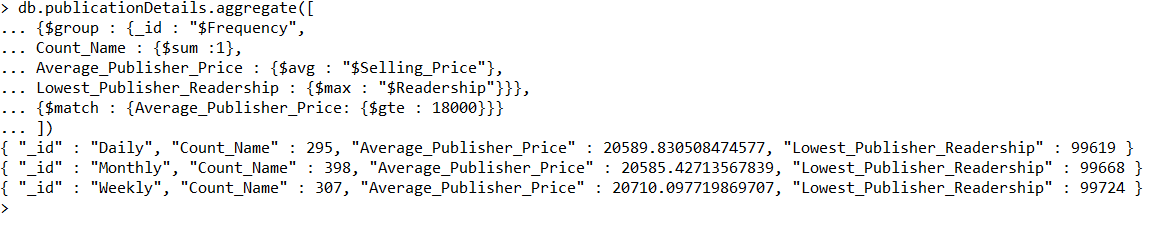
Count\_Name : {$sum :1},

Average\_Publisher\_Price : {$avg : "$Selling\_Price"},

Lowest\_Publisher\_Readership : {$max : "$Readership"}}},

{$match : {Average\_Publisher\_Price: {$gte : 18000}}}

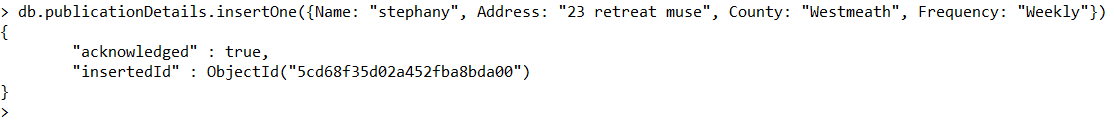
])



# Queries of CRUD:

**Question: 31 Add a document in publisher collection detail that has only following detail name, address,county and frequency.**

db.publicationDetails.insertOne({Name: "stephany", Address: "23 retreat muse", County: "Westmeath", Frequency: "Weekly"})



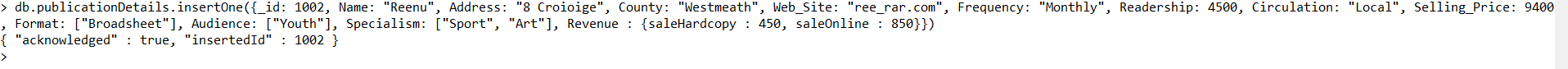
**//How many documents in collection after insert//**

db.publicationDetails.count()



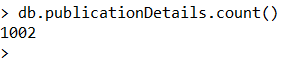
**Question: 32 Add a publisher with the full set of details that way it has a value under every key value pair in the original file.**

db.publicationDetails.insertOne({\_id: 1002, Name: "Reenu", Address: "8 Croioige", County: "Westmeath", Web\_Site: "ree\_rar.com", Frequency: "Monthly", Readership: 4500, Circulation: "Local", Selling\_Price: 9400, Format: ["Broadsheet"], Audience: ["Youth"], Specialism: ["Sport", "Art"], Revenue : {saleHardcopy : 450, saleOnline : 850}})



**//How many documents is in collection after insert//**

db.publicationDetails.count()



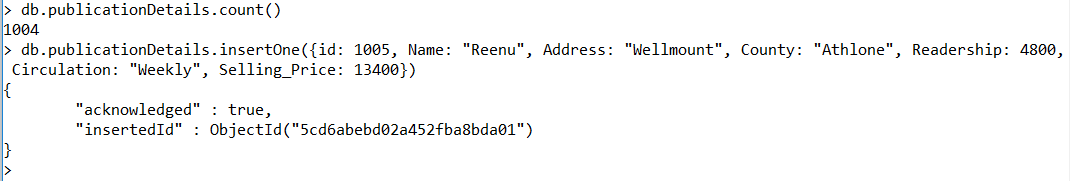
**//Find the document just inserted//**

db.publicationDetails.find({\_id: 1002}).pretty()



**Question 33: Add a document in publisher collection ensuring, that insert includes readership, circulation and selling price.**

db.publicationDetails.insertOne({id: 1005, Name: "Reenu", Address: "Wellmount", County: "Athlone", Readership: 4800, Circulation: "Weekly", Selling\_Price: 13400})



**Question 34: Add three documents in publisher collection ensuring that audience, format and specialism is being recorded.**

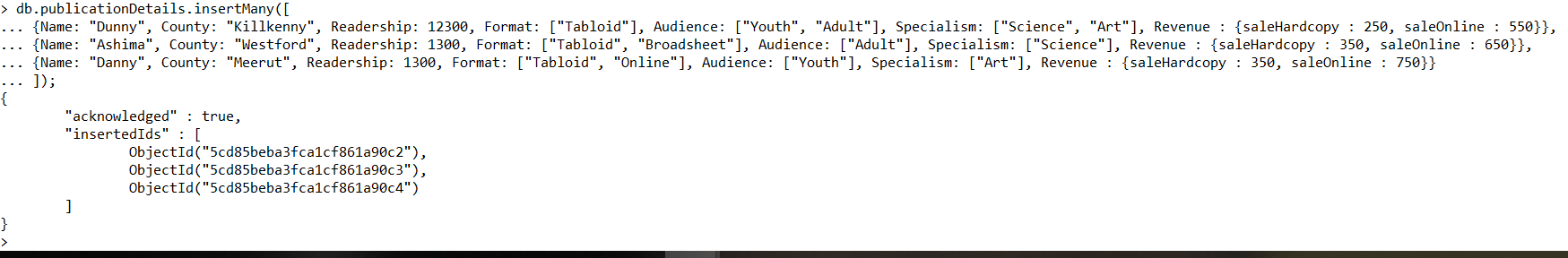
db.publicationDetails.insertMany([

{Name: "Dunny", County: "Killkenny", Readership: 12300, Format: ["Tabloid"], Audience: ["Youth", "Adult"], Specialism: ["Science", "Art"], Revenue : {saleHardcopy : 250, saleOnline : 550}},

{Name: "Ashima", County: "Westford", Readership: 1300, Format: ["Tabloid", "Broadsheet"], Audience: ["Adult"], Specialism: ["Science"], Revenue : {saleHardcopy : 350, saleOnline : 650}},

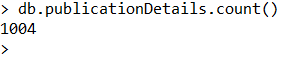
{Name: "Danny", County: "Meerut", Readership: 1300, Format: ["Tabloid", "Online"], Audience: ["Youth"], Specialism: ["Art"], Revenue : {saleHardcopy : 350, saleOnline : 750}}

]);



**//How many documents in collection after insert//**

db.publicationDetails.count()



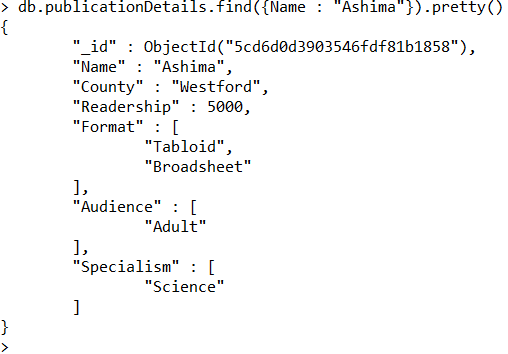
**Question 35: Modify the readership count to 5000 for first publisher encountered named Ashima.**

db.publicationDetails.updateOne({Name: "Ashima"}, {$set: {Readership : 5000}})



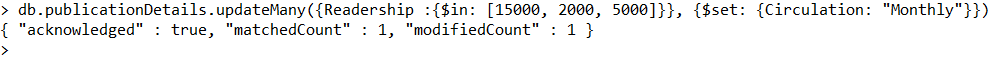
**//Confirm the changes to readership for given publisher//**

db.publicationDetails.find({Name : "Ashima"}).pretty()



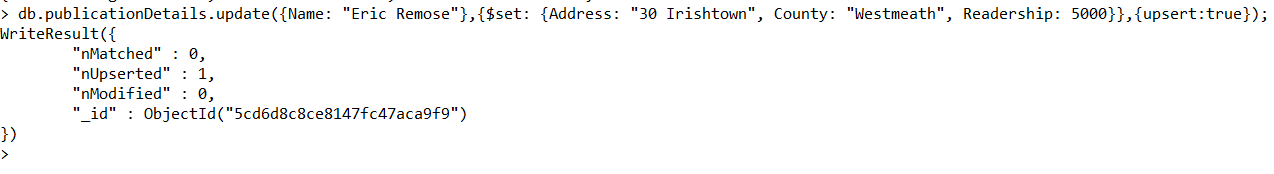
**Question :36 Update the circulation for the publisher who has readership any of following: 15000, 2000 and 5000.**

db.publicationDetails.updateMany({Readership :{$in: [15000, 2000, 5000]}}, {$set: {Circulation: "Monthly"}})



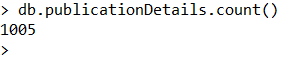
**Question :37 Adjust the address to 30 Irishtown, county to Westmeath and readership to 5000 for publisher name "Eric Remose" and insert the publisher if no matching found in collection.**

db.publicationDetails.update({Name: "Eric Remose"},{$set: {Address: "30 Irishtown", County: "Westmeath", Readership: 5000}},{upsert:true});



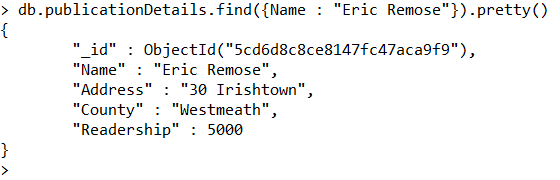
**//Count the document to ensure the last update or insert//**

db.publicationDetails.count()



**//Confirm the update//**

db.publicationDetails.find({Name : "Eric Remose"}).pretty()



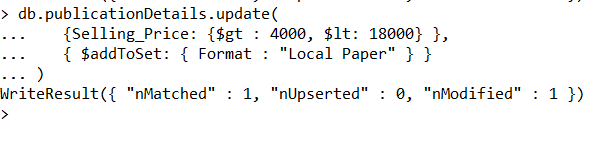
**Question :38 Add the "Local Paper" in formats for first encountered publisher whose selling price are greater than 4000 and less than 18000.**

db.publicationDetails.update(

{Selling\_Price: {$gt : 4000, $lt: 18000} },

{ $addToSet: { Format : "Local Paper" } }

)

**//Display the document in which new array field is inserted//**

db.publicationDetails.find({Selling\_Price: {$gt : 4000, $lt: 18000}}).limit(1).pretty()



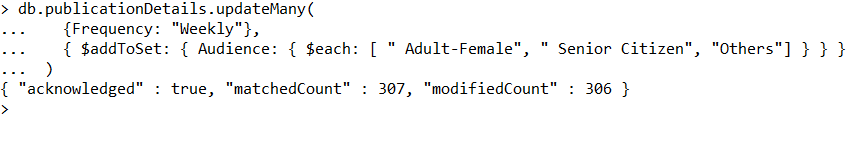
**Question :39 Add the Adult-Female, Senior Citizen and Others in audience list (without repetition) for each publisher who has frequency recorded as "Weekly".**

db.publicationDetails.updateMany(

{Frequency: "Weekly"},

{ $addToSet: { Audience: { $each: [ " Adult-Female", " Senior Citizen", "Others"] } } }

)



**//Display the document in which new array field is inserted//**

db.publicationDetails.find({Frequency: "Weekly"}).limit(1).pretty()



**Question: 40 Delete first the first document encountered from publisher collection where county is Westmeath.**

db.publicationDetails.remove({County: "Westmeath"})



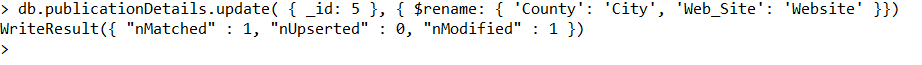
**//Confirm how many document are in collection afterwards//**

db.publicationDetails.count()



**Question: 41 Rename the County and Web\_Site Field to City and Website for document whose id is 5 in publisher collection.**

db.publicationDetails.update( { \_id: 5 }, { $rename: { 'County': 'City', 'Web\_Site': 'Website' }})



**//Confirm the changes for id 5//**

db.publicationDetails.find({ \_id: 5 }).pretty()



db.publicationDetails.update( { \_id: 5 }, { $rename: { 'City': 'County', 'Website': 'Web\_Site' }})

**Question: 42 Remove the Young- Female and Others from Audience and Local-Paper from format for all document of collection array.**

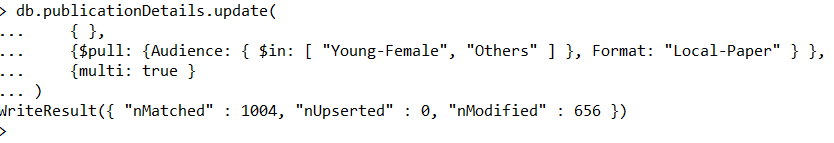
db.publicationDetails.update(

{ },

{$pull: {Audience: {$in: [ "Young-Female", "Others"]}, Format: "Local-Paper" } },

{multi: true}

)



# You Tube Screencast links:

1. <https://youtu.be/8G_YpjZDm6Q>
2. <https://youtu.be/F4cHTKN8Bc4>

# References:

1.https://docs.mongodb.com/manual/reference/operator/query/#query-selectors

2.https://docs.mongodb.com/manual/reference/operator/query/regex/index.html

3. <https://www.tutorialspoint.com/mongodb/mongodb_overview.htm>