**Assignment 1**

1. Create a database with the name ‘Movie’.
2. A ‘Film’ is a collection of documents with the following fields:
   1. Film Id
   2. Title of the film
   3. Year of release
   4. Genre / Category (like adventure, action, sci-fi, romantic etc.) A film can belong to more than one genre.
   5. Actors (First name and Last name)

A film can have more than one actor.

* 1. Director (First name and Last name)

A film can have more than one director.

* 1. Release details (It consists of places of release, dates of release and rating of the film.)

1. An ‘Actor’ is a collection of documents with the following fields:
   1. Actor Id
   2. First name
   3. Last Name
   4. Address (Street, City, State, Country, Pin-code)
   5. Contact Details (Email Id and Phone No)
   6. Age of an actor.

Queries:

1. Insert at least 10 documents in the collection Film –
   1. Insert at least one document with film belonging to two genres.
   2. Insert at least one document with film that is released at more than one place and on two different dates.
   3. Insert at least three documents with the films released in the same year.
   4. Insert at least two documents with the films directed by one director.
   5. Insert at least two documents with films those are acted by a pair ‘Madhuri Dixit’ and ‘Shahrukh Khan’.
2. Insert at least 10 documents in the collection Actor.

Make sure, you are inserting the names of actors who have acted in films, given in the ‘Film’ collection.

1. Display all the documents inserted in both the collections.
2. Add a value to the rating of the film whose title starts with ‘T’.
3. Add an actor named " " in the ‘Actor’ collection. Also add the details of the film in ‘Film’ collection in which this actor has acted in.
4. Delete the film " ".
5. Delete an actor named " ".
6. Delete all actors from an ‘Actor’ collection who have age greater than “ ”
7. Update the actor’s address where Actor Id is “ ”. 10.Update the genre of the film directed by “ ”.

X

**Assignment 2:**

1. Create a database with name ‘Company’.
2. An ‘Employee’ is a collection of documents with the following fields:
   1. Employee ID
   2. First Name
   3. Last Name
   4. Email
   5. Phone No.
   6. Address (House No, Street, City, State, Country, Pin-code)
   7. Salary
   8. Designation
   9. Experience
   10. Date of Joining
   11. Birthdate
3. A ‘Transaction’ is a collection of documents with the following fields:
   1. Transaction Id,
   2. Transaction Date
   3. Name (First Name of employee who processed the transaction)
   4. Transaction Details (Item Id, Item Name, Quantity, Price)
   5. Payment (Type of Payment (Debit/Credit/Cash), Total amount paid, Payment Successful)
   6. Remark (Remark field can be empty.) Queries:
4. Insert at least 5 documents in ‘Employee’ collection.
5. Insert multiple documents (at least 10) into the ‘Transaction’ collection by passing an array of documents to the db.collection.insert () method.
6. Display all the documents of both the collections in a formatted manner.
7. Update salary of all employees by giving an increment of Rs. 4000.
8. Update the remark for transaction id 201.
9. Update designation of an employee named “\_ ” from supervisor to manager.
10. Update designation of an employee having Employee Id as .
11. Change the address of an employee having Employee Id as .
12. Delete transaction made by “ ” employee on the given date. 10.Delete all the employees whose first name starts with ‘K’.

X

**Assignment 3:**

This assignment is based on ‘Movie’ database having collections ‘Film’ and ‘Actor’.

**Prerequisite**: Read MongoDB Aggregate framework before executing the following assignments.

Note: It is expected that student should fill in the data relevant to the queries given in the assignment. The result set should not be empty.

1. Find the titles of all the films starting with the letter ‘R’ released during the year 2009 and 2011.
2. Find the list of films acted by an actor " ".
3. Find all the films released in 90s.
4. Find all films belonging to “Adventure” and “Thriller” genre.
5. Find all the films having ‘A’ rating.
6. Arrange the film names in ascending order and release year should be in descending order.
7. Sort the actors in ascending order according to their age.
8. Find movies that are comedies or dramas and are released after 2013.
9. Show the latest 2 films acted by an actor “ ”.
10. List the titles of films acted by actors “ ” and “ ”. 11.Retrieve films with an actor living in Spain.

12.Retrieve films with actor details.

Note: Similarly, additional queries can be executed based on these collections for practice.

X

Assignment 4:

This assignment is based on ‘Company’ database having collections ‘Employee’ and ‘Transaction’.

**Prerequisite**: Read MongoDB Aggregate framework before executing the following assignments.

Note: It is expected that student should fill in the data relevant to the queries given in the assignment. The result set should not be empty.

1. Find employees having designation as either ‘manager’ or ‘floor supervisor’.
2. Find an employee whose name ends with " " and print the output in json format.
3. Display the name of an employee whose salary is greater than using a MongoDB cursor.
4. Sort the employees in the descending order of their designation.
5. Count the total number of employees in a collection.
6. Calculate the sum of total amount paid for all the transaction documents.
7. Calculate the sum of total amount paid for each payment type.
8. Find the transaction id of the latest transaction.
9. Find designation of employees who have made transaction of amount greater than Rs. 500.
10. Find the total quantity of a particular item sold using Map Reduce.

X

Prepared by:

Mrs. Chitra Alavani, Kaveri College of Arts, Science and Commerce. Mrs. Rasika Rahalkar, MES’ Abasaheb Garware College.

***Ahemadnagar Jilha Maratha Prasarak Samaj’s***

## Rajarshi Shahu Mahavidyalaya, Deolali Pravara

**Class: M.Sc(Computer Science)-I Assignment no: 1**

**Name: Sign:**

**Title: Movie Database Date:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

C:\Users\user>cd C:\Program Files\MongoDB\Server\4.2\bin C:\Program Files\MongoDB\Server\4.2\bin>mongo

* show dbs
* use Movie

switched to db Movie

# Create a database with the name „Movie‟.

1. **A „Film‟ is a collection of documents with the following fields:**
   1. **Film Id**
   2. **Title of the film**
   3. **Year of release**
   4. **Genre / Category (like adventure, action, sci-fi, romantic etc.)A film can belong to more than one genre.**
   5. **Actors (First name and Last name)**

**A film can have more than one actor.**

* 1. **Director (First name and Last name)**

**A film can have more than one director.**

* 1. **Release details (It consists of places of release, dates of release and rating of the film.)**

# An „Actor‟ is a collection of documents with the following fields:

* 1. **Actor Id**
  2. **First name**
  3. **Last Name**
  4. **Address (Street, City, State, Country, Pin-code)**
  5. **Contact Details (Email Id and Phone No)**
  6. **Age of an actor.**

# Queries:

1. **Insert at least 10 documents in the collection Film –**
   1. **Insert at least one document with film belonging to two genres.**
   2. **Insert at least one document with film that is released at more than one place and on two different dates.**
   3. **Insert at least three documents with the films released in the same year.**
   4. **Insert at least two documents with the films directed by one director.**
   5. **Insert at least two documents with films those are acted by a pair**

**„Madhuri Dixit‟ and „Shahrukh Khan‟.**

* db.Film.insert({\_id:12,TitleofFilm:"The Avengers",YearofRelease:2012,Genre:["Action","Adventure","SciFi"],Actor:{FN

:["Robert","Chris","Chris"],LN:["Downey","Evans","Hemsworth"]},Director:{FN

:["Joss"],LN:["Whedon"]},ReleaseDetails:{Places:["USA","India"],Dates:[new

Date("2012-04-22"),new Date("2012-04-27")],Rating:"PG"}})

WriteResult({ "nInserted" : 1 })

* db.Film.insert({\_id:13,TitleofFilm:"Real Steel",YearofRelease:2011,Genre:["Action","SciFi"],Actor:{FN:["Hugh"],LN:["

Jackman"]},Director:{FN:["Shawn"],LN:["Levy"]},ReleaseDetails:{Places:["USA

"],Dates:new Date("2011-10-07"),Rating:"PG"}}) WriteResult({ "nInserted" : 1 })

>

db.Film.insert({\_id:14,TitleofFilm:"Housefull3",YearofRelease:2016,Genre:[" Comedy","Drama"],Actor:{FN:["Akshay","Abhishek"],LN:["Kumar","Bachhan"]},Di

rector:{FN:["Sajid","Farhad"],LN:["Khan","Samji"]},ReleaseDetails:{Places:[

"India"],Dates:new Date("2016-06-03"),Rating:"N"}}) WriteResult({ "nInserted" : 1 })

* db.Film.insert({\_id:15,TitleofFilm:"Dil To Pagal Hai",YearofRelease:1997,Genre:["Comedy","Drama"],Actor:{FN:["Shahrukh","Mad huri"],LN:["Khan","Dixit"]},Director:{FN:["Yash"],LN:["Chopra"]},ReleaseDet

ails:{Places:["India"],Dates:new Date("1997-10-30"),Rating:"N"}}) WriteResult({ "nInserted" : 1 })

>

db.Film.insert({\_id:16,TitleofFilm:"Devdas",YearofRelease:2002,Genre:["Roma nce","Drama"],Actor:{FN:["Shahrukh","Madhuri"],LN:["Khan","Dixit"]},Directo

r:{FN:["Sanjay"],LN:["Bhansali"]},ReleaseDetails:{Places:["India"],Dates:ne w Date("2002-07-12"),Rating:"A"}})

WriteResult({ "nInserted" : 1 })

>

db.Film.insert({\_id:17,TitleofFilm:"Inception",YearofRelease:2010,Genre:["A dventure","Thriller"],Actor:{FN:["Leonardo","Tom"],LN:["DiCaprio","Hardy"]}

,Director:{FN:["Christopher"],LN:["Nolan"]},ReleaseDetails:{Places:["USA"], Dates:new Date("2010-07-16"),Rating:"PG"}})

WriteResult({ "nInserted" : 1 })

* db.Film.insert({\_id:18,TitleofFilm:"Mission Impossible4",YearofRelease:2011,Genre:["Adventure","Action","Thriller"],Act or:{FN:["Tom","Jeremy"],LN:["Cruise","Renner"]},Director:{FN:["Brad"],LN:[" Bird"]},ReleaseDetails:{Places:["Dubai","India"],Dates:[new Date("2011-12- 07"),new Date("2011-12-16")],Rating:"PG"}})

WriteResult({ "nInserted" : 1 })

* db.Film.insert({\_id:19,TitleofFilm:"Chiller Party",YearofRelease:2011,Genre:["Comedy","Drama"],Actor:{FN:["Naman","Irfa n"],LN:["Jain","Khan"]},Director:{FN:["Vikas","Nitesh"],LN:["Bahl","Tiwari"

]},ReleaseDetails:{Places:["India"],Dates:new Date("2011-07- 08"),Rating:"A"}})

WriteResult({ "nInserted" : 1 })

* db.Film.insert({\_id:20,TitleofFilm:"Angry Birds",YearofRelease:2016,Genre:["Comedy","Animation","Drama"],Actor:{FN:[" Jason","Josh"],LN:["Sudeikis","Gad"]},Director:{FN:["Clay","Fergal"],LN:["K aytis","Reilly"]},ReleaseDetails:{Places:["India","USA"],Dates:[new Date("2016-05-20"),new Date("2016-05-28")],Rating:"PG"}})

WriteResult({ "nInserted" : 1 })

>

db.Film.insert({\_id:21,TitleofFilm:"RaOne",YearofRelease:2011,Genre:["Actio n","SciFi"],Actor:{FN:["Shahrukh","Kareena"],LN:["Khan","Kapoor"]},Director

:{FN:["Anubhav"],LN:["Sinha"]},ReleaseDetails:{Places:["India"],Dates:[new Date("2011-10-26")],Rating:"N"}})

WriteResult({ "nInserted" : 1 })

# Insert at least 10 documents in the collection Actor. Make sure, you are inserting the names of actors who have acted in films, given in the „Film‟ collection.

>

db.Actor.insert({\_id:1,FN:"Robert",LN:"Downey",Address:{Street:1311,City:"M alibu",State:"California",Country:"Los [Angeles",Pincode:413716},Contact:{EmailId:"robert123@gmail.com",Phone:98765](mailto:robert123@gmail.com) 43210},Age:53})

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:2,FN:"Shahrukh",LN:"Khan",Address:{Street:1311,City:"M umbai",State:"Maharashtra",Country:"India",Pincode:413716},Contact:{EmailId

[:"srk123@gmail.com",Phone:9876543210},Age:49})](mailto:srk123@gmail.com) WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:3,FN:"Tom",LN:"Cruise",Address:{Street:1321,City:"Spai n",State:"California",Country:"Los [Angeles",Pincode:413716},Contact:{EmailId:"tomcruise123@gmail.com",Phone:98](mailto:tomcruise123@gmail.com) 76543210},Age:57})

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:4,FN:"Kareena",LN:"Kapoor",Address:{Street:1311,City:" Mumbai",State:"Maharashtra",Country:"India",Pincode:413716},Contact:{EmailI [d:"kareena123@gmail.com",Phone:9876543210},Age:39})](mailto:kareena123@gmail.com)

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:5,FN:"Madhuri",LN:"Dixit",Address:{Street:1311,City:"M umbai",State:"Maharashtra",Country:"India",Pincode:413716},Contact:{EmailId

[:"madhuri123@gmail.com",Phone:9876543210},Age:52})](mailto:madhuri123@gmail.com) WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:6,FN:"Akshay",LN:"Kumar",Address:{Street:1311,City:"Mu mbai",State:"Maharashtra",Country:"India",Pincode:413716},Contact:{EmailId: ["akshay123@gmail.com",Phone:9876543210},Age:52})](mailto:akshay123@gmail.com)

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:7,FN:"Abhishek",LN:"Bachhan",Address:{Street:1311,City

:"Mumbai",State:"Maharashtra",Country:"India",Pincode:413716},Contact:{Emai [lId:"abhishek123@gmail.com",Phone:9876543210},Age:43})](mailto:abhishek123@gmail.com)

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:8,FN:"Hugh",LN:"Jackman",Address:{Street:1321,City:"Sp ain",State:"California",Country:"Los [Angeles",Pincode:413716},Contact:{EmailId:"jackman123@gmail.com",Phone:9876](mailto:jackman123@gmail.com) 543210},Age:51})

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:9,FN:"Chris",LN:"Hemsworth",Address:{Street:1321,City: "Spain",State:"California",Country:"Los [Angeles",Pincode:413716},Contact:{EmailId:"chrish123@gmail.com",Phone:98765](mailto:chrish123@gmail.com) 43210},Age:36})

WriteResult({ "nInserted" : 1 })

>

db.Actor.insert({\_id:10,FN:"Leonardo",LN:"DiCaprio",Address:{Street:1321,Ci ty:"Spain",State:"California",Country:"Los [Angeles",Pincode:413716},Contact:{EmailId:"leo123@gmail.com",Phone:98765432](mailto:leo123@gmail.com) 10},Age:44})

WriteResult({ "nInserted" : 1 })

# Display all documents of collection

* db.Film.find()

{ "\_id" : 12, "TitleofFilm" : "The Avengers", "YearofRelease" : 2012, "Genre" : [ "Action", "Adventure", "SciFi" ], "Actor" : { "FN" : [

"Robert", "Chris", "Chris" ], "LN" : [ "Downey", "Evans", "Hemsworth" ] },

"Director" : { "FN" : [ "Joss" ], "LN" : [ "Whedon" ] }, "ReleaseDetails" :

{ "Places" : [ "USA", "India" ], "Dates" : [ ISODate("2012-04-

22T00:00:00Z"), ISODate("2012-04-27T00:00:00Z") ], "Rating" : "PG" } }

{ "\_id" : 13, "TitleofFilm" : "Real Steel", "YearofRelease" : 2011, "Genre"

: [ "Action", "SciFi" ], "Actor" : { "FN" : [ "Hugh" ], "LN" : [ "Jackman"

] }, "Director" : { "FN" : [ "Shawn" ], "LN" : [ "Levy" ] },

"ReleaseDetails" : { "Places" : [ "USA" ], "Dates" : ISODate("2011-10- 07T00:00:00Z"), "Rating" : "PG" } }

{ "\_id" : 14, "TitleofFilm" : "Housefull3", "YearofRelease" : 2016, "Genre"

: [ "Comedy", "Drama" ], "Actor" : { "FN" : [ "Akshay", "Abhishek" ], "LN"

: [ "Kumar", "Bachhan" ] }, "Director" : { "FN" : [ "Sajid", "Farhad" ],

"LN" : [ "Khan", "Samji" ] }, "ReleaseDetails" : { "Places" : [ "India" ],

"Dates" : ISODate("2016-06-03T00:00:00Z"), "Rating" : "N" } }

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai", "YearofRelease" : 1997, "Genre" : [ "Comedy", "Drama" ], "Actor" : { "FN" : [ "Shahrukh", "Madhuri"

], "LN" : [ "Khan", "Dixit" ] }, "Director" : { "FN" : [ "Yash" ], "LN" : [

"Chopra" ] }, "ReleaseDetails" : { "Places" : [ "India" ], "Dates" :

ISODate("1997-10-30T00:00:00Z"), "Rating" : "N" } }

{ "\_id" : 16, "TitleofFilm" : "Devdas", "YearofRelease" : 2002, "Genre" : [

"Romance", "Drama" ], "Actor" : { "FN" : [ "Shahrukh", "Madhuri" ], "LN" :

[ "Khan", "Dixit" ] }, "Director" : { "FN" : [ "Sanjay" ], "LN" : [

"Bhansali" ] }, "ReleaseDetails" : { "Places" : [ "India" ], "Dates" :

ISODate("2002-07-12T00:00:00Z"), "Rating" : "A" } }

{ "\_id" : 17, "TitleofFilm" : "Inception", "YearofRelease" : 2010, "Genre"

: [ "Adventure", "Thriller" ], "Actor" : { "FN" : [ "Leonardo", "Tom" ],

"LN" : [ "DiCaprio", "Hardy" ] }, "Director" : { "FN" : [ "Christopher" ],

"LN" : [ "Nolan" ] }, "ReleaseDetails" : { "Places" : [ "USA" ], "Dates" : ISODate("2010-07-16T00:00:00Z"), "Rating" : "PG" } }

{ "\_id" : 18, "TitleofFilm" : "Mission Impossible4", "YearofRelease" : 2011, "Genre" : [ "Adventure", "Action", "Thriller" ], "Actor" : { "FN" : [

"Tom", "Jeremy" ], "LN" : [ "Cruise", "Renner" ] }, "Director" : { "FN" : [

"Brad" ], "LN" : [ "Bird" ] }, "ReleaseDetails" : { "Places" : [ "Dubai",

"India" ], "Dates" : [ ISODate("2011-12-07T00:00:00Z"), ISODate("2011-12- 16T00:00:00Z") ], "Rating" : "PG" } }

{ "\_id" : 19, "TitleofFilm" : "Chiller Party", "YearofRelease" : 2011, "Genre" : [ "Comedy", "Drama" ], "Actor" : { "FN" : [ "Naman", "Irfan" ],

"LN" : [ "Jain", "Khan" ] }, "Director" : { "FN" : [ "Vikas", "Nitesh" ],

"LN" : [ "Bahl", "Tiwari" ] }, "ReleaseDetails" : { "Places" : [ "India" ],

"Dates" : ISODate("2011-07-08T00:00:00Z"), "Rating" : "A" } }

{ "\_id" : 20, "TitleofFilm" : "Angry Birds", "YearofRelease" : 2016, "Genre" : [ "Comedy", "Animation", "Drama" ], "Actor" : { "FN" : [ "Jason",

"Josh" ], "LN": [ "Sudeikis", "Gad" ] }, "Director" : { "FN" : [ "Clay",

"Fergal" ], "LN" : [ "Kaytis", "Reilly" ] }, "ReleaseDetails" : { "Places"

: [ "India", "USA" ], "Dates" : [ ISODate("2016-05-20T00:00:00Z"),

ISODate("2016-05-28T00:00:00Z") ], "Rating" : "PG" } }

{ "\_id" : 21, "TitleofFilm" : "RaOne", "YearofRelease" : 2011, "Genre" : [

"Action", "SciFi" ], "Actor" : { "FN" : [ "Shahrukh", "Kareena" ], "LN" : [

"Khan", "Kapoor" ] }, "Director" : { "FN" : [ "Anubhav" ], "LN" : [ "Sinha"

] }, "ReleaseDetails" : { "Places" : [ "India" ], "Dates" : [

ISODate("2011-10-26T00:00:00Z") ], "Rating" : "N" } }

* db.Actor.find()

{ "\_id" : 1, "FN" : "Robert", "LN" : "Downey", "Address" : { "Street" : 1311, "City" : "Malibu", "State" : "California", "Country" : "Los Angeles", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["robe](mailto:robert123@gmail.com)[rt123@gmail.com",](mailto:rt123@gmail.com) "Phone" : 9876543210 }, "Age" : 53 }

{ "\_id" : 2, "FN" : "Shahrukh", "LN" : "Khan", "Address" : { "Street" :

1311, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["srk123@gmail.com",](mailto:srk123@gmail.com) "Phone"

: 9876543210 }, "Age" : 49 }

{ "\_id" : 3, "FN" : "Tom", "LN" : "Cruise", "Address" : { "Street" : 1321,

"City" : "Spain", "State" : "California", "Country" : "Los Angeles", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["tomcruise123@gmail.com",](mailto:tomcruise123@gmail.com) "Phone" : 9876543210 }, "Age" : 57 }

{ "\_id" : 4, "FN" : "Kareena", "LN" : "Kapoor", "Address" : { "Street" :

1311, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["kareena123@gmail.com",](mailto:kareena123@gmail.com) "Phone" : 9876543210 }, "Age" : 39 }

{ "\_id" : 5, "FN" : "Madhuri", "LN" : "Dixit", "Address" : { "Street" :

1311, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["madhuri123@gmail.com"](mailto:madhuri123@gmail.com), "Phone" : 9876543210 }, "Age" : 52 }

{ "\_id" : 6, "FN" : "Akshay", "LN" : "Kumar", "Address" : { "Street" :

1311, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["akshay123@gmail.com",](mailto:akshay123@gmail.com) "Phone" : 9876543210 }, "Age" : 52 }

{ "\_id" : 7, "FN" : "Abhishek", "LN" : "Bachhan", "Address" : { "Street" :

1311, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["abhishek123@gmail.com",](mailto:abhishek123@gmail.com) "Phone" : 9876543210 }, "Age" : 43 }

{ "\_id" : 8, "FN" : "Hugh", "LN" : "Jackman", "Address" : { "Street" : 1321, "City" : "Spain", "State" : "California", "Country" : "Los Angeles", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["jackman123@gmail.com",](mailto:jackman123@gmail.com) "Phone" : 9876543210 }, "Age" : 51 }

{ "\_id" : 9, "FN" : "Chris", "LN" : "Hemsworth", "Address" : { "Street" : 1321, "City" : "Spain", "State" : "California", "Country" : "Los Angeles", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["chrish123@gmail.com",](mailto:chrish123@gmail.com) "Phone" : 9876543210 }, "Age" : 36 }

{ "\_id" : 10, "FN" : "Leonardo", "LN" : "DiCaprio", "Address" : { "Street"

: 1321, "City" : "Spain", "State" : "California", "Country" : "Los Angeles", "Pincode" : 413716 }, "Contact" : { "EmailId" : ["leo123@gmail.com",](mailto:leo123@gmail.com) "Phone" : 9876543210 }, "Age" : 44 }

# Add a value to the Rating of the film whose title starts with „T‟

>

db.Film.update({"TitleofFilm":{$regex:/^T/}},{$set:{"ReleaseDetails.Rating"

:"A"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Delete the film „Chiller Party‟

* db.Film.remove({TitleofFilm:"Chiller Party"}) WriteResult({ "nRemoved" : 1 })

# Delete an actor named „Kareena‟

* db.Actor.remove({FN:"Kareena"}) WriteResult({ "nRemoved" : 1 })

# Delete all actors from „Actor collection who have age greater than „55‟

* db.Actor.remove({Age:{$gt:55}}) WriteResult({ "nRemoved" : 1 })

# Update the actor‟s address where ActorId is „9‟

>

db.Actor.update({\_id:9},{$set:{Address:{Street:1321,City:"London",State:"Ca lifornia",Country:"Los Angeles",Pincode:413716}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Update the genre of the film directed by „Christopher Nolan‟

>

db.Film.update({"Director.FN":"Christopher","Director.LN":"Nolan"},{$set:{" Genre":["Mystery","Adventure","Thriller"]}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

***Ahemadnagar Jilha Maratha Prasarak Samaj’s***

## Rajarshi Shahu Mahavidyalaya, Deolali Pravara

**Class: M.Sc(Computer Science)-I Assignment no: 3**

**Name: Sign:**

**Title: Movie Database Date:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

C:\Users\user>cd C:\Program Files\MongoDB\Server\4.2\bin C:\Program Files\MongoDB\Server\4.2\bin>mongo

* show dbs
* use Movie

switched to db Movie

# This assignment is based on „Movie‟ database having collections „Film‟ and „Actor‟. Using MongoDB Aggregate framework

1. **Find the titles of all films starting with the letter „D‟ And released during year 1996 and 2003**

>

db.Film.find({"TitleofFilm":{$regex:/^D/},"YearofRelease":{$gt:1996,$lt:200 3}},{"TitleofFilm":1})

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai" }

{ "\_id" : 16, "TitleofFilm" : "Devdas" }

# Find the list of films acted by an actor “Shahrukh Khan”

* db.Film.find({"Actor.FN":"Shahrukh","Actor.LN":"Khan"},{"TitleofFilm":1})

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai" }

{ "\_id" : 16, "TitleofFilm" : "Devdas" }

{ "\_id" : 21, "TitleofFilm" : "RaOne" }

# Find all the films released in 90s

* db.Film.find({YearofRelease:{$lt:2000}})

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai", "YearofRelease" : 1997, "Genre" : [ "Comedy", "Drama" ], "Actor" : { "FN" : [ "Shahrukh", "Madhuri"

], "LN" : [ "Khan", "Dixit" ] }, "Director" : { "FN" : [ "Yash" ], "LN" : [

"Chopra" ] }, "ReleaseDetails" : { "Places" : [ "India" ], "Dates" :

ISODate("1997-10-30T00:00:00Z"), "Rating" : "N" } }

# Find all films belonging to „Adventure‟ and „Thriller‟ genre

* db.Film.find({"Genre":"Adventure","Genre":"Thriller"})

{ "\_id" : 17, "TitleofFilm" : "Inception", "YearofRelease" : 2010, "Genre"

: [ "Adventure", "Thriller" ], "Actor" : { "FN" : [ "Leonardo", "Tom" ],

"LN" : [ "DiCaprio", "Hardy" ] }, "Director" : { "FN" : [ "Christopher" ],

"LN" : [ "Nolan" ] }, "ReleaseDetails" : { "Places" : [ "USA" ], "Dates" : ISODate("2010-07-16T00:00:00Z"), "Rating" : "PG" } }

{ "\_id" : 18, "TitleofFilm" : "Mission Impossible4", "YearofRelease" : 2011, "Genre" : [ "Adventure", "Action", "Thriller" ], "Actor" : { "FN" : [

"Tom", "Jeremy" ], "LN" : [ "Cruise", "Renner" ] }, "Director" : { "FN" : [

"Brad" ], "LN" : [ "Bird" ] }, "ReleaseDetails" : { "Places" : [ "Dubai",

"India" ], "Dates" : [ ISODate("2011-12-07T00:00:00Z"), ISODate("2011-12- 16T00:00:00Z") ], "Rating" : "PG" } }

# Find all the films having „A‟ rating

* db.Film.find({"ReleaseDetails.Rating":"A"})

{ "\_id" : 19, "TitleofFilm" : "Chiller Party", "YearofRelease" : 2011, "Genre" : [ "Comedy", "Drama" ], "Actor" : { "FN" : [ "Naman", "Irfan" ],

"LN" : [ "Jain", "Khan" ] }, "Director" : { "FN" : [ "Vikas", "Nitesh" ],

"LN" : [ "Bahl", "Tiwari" ] }, "ReleaseDetails" : { "Places" : [ "India" ],

"Dates" : ISODate("2011-07-08T00:00:00Z"), "Rating" : "A" } }

# Arrange the film names in ascending order and release year should be in descending order

>

db.Film.find({},{"TitleofFilm":1,"YearofRelease":1}).sort({"TitleofFilm":1, "YearofRelease":-1})

{ "\_id" : 20, "TitleofFilm" : "Angry Birds", "YearofRelease" : 2016 }

{ "\_id" : 19, "TitleofFilm" : "Chiller Party", "YearofRelease" : 2011 }

{ "\_id" : 16, "TitleofFilm" : "Devdas", "YearofRelease" : 2002 }

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai", "YearofRelease" : 1997 }

{ "\_id" : 14, "TitleofFilm" : "Housefull3", "YearofRelease" : 2016 }

{ "\_id" : 17, "TitleofFilm" : "Inception", "YearofRelease" : 2010 }

{ "\_id" : 18, "TitleofFilm" : "Mission Impossible4", "YearofRelease" : 2011

}

{ "\_id" : 21, "TitleofFilm" : "RaOne", "YearofRelease" : 2011 }

{ "\_id" : 13, "TitleofFilm" : "Real Steel", "YearofRelease" : 2011 }

{ "\_id" : 12, "TitleofFilm" : "The Avengers", "YearofRelease" : 2012 }

# Sort the actors in ascending order of their age

* db.Actor.find({},{"FN":1,"Age":1}).sort({"Age":-1})

{ "\_id" : 3, "FN" : "Tom", "Age" : 57 }

{ "\_id" : 1, "FN" : "Robert", "Age" : 53 }

{ "\_id" : 5, "FN" : "Madhuri", "Age" : 52 }

{ "\_id" : 6, "FN" : "Akshay", "Age" : 52 }

{ "\_id" : 8, "FN" : "Hugh", "Age" : 51 }

{ "\_id" : 2, "FN" : "Shahrukh", "Age" : 49 }

{ "\_id" : 10, "FN" : "Leonardo", "Age" : 44 }

{ "\_id" : 7, "FN" : "Abhishek", "Age" : 43 }

{ "\_id" : 4, "FN" : "Kareena", "Age" : 39 }

{ "\_id" : 9, "FN" : "Chris", "Age" : 36 }

# Find movies that are „Adventure‟ or „Animation‟ and are released after 2010

>

db.Film.find({$and:[{$or:[{"Genre":"Adventure"},{"Genre":"Animation"}]},{"Y

earofRelease":{$gt:2010}}]},{"TitleofFilm":1,"Genre":1,"YearofRelease":1})

{ "\_id" : 12, "TitleofFilm" : "The Avengers", "YearofRelease" : 2012, "Genre" : [ "Action", "Adventure", "SciFi" ] }

{ "\_id" : 18, "TitleofFilm" : "Mission Impossible4", "YearofRelease" : 2011, "Genre" : [ "Adventure", "Action", "Thriller" ] }

{ "\_id" : 20, "TitleofFilm" : "Angry Birds", "YearofRelease" : 2016, "Genre" : [ "Comedy", "Animation", "Drama" ] }

# Show the latest 2 films acted by an actor „Shahrukh Khan‟

>

db.Film.find({"Actor.FN":"Shahrukh","Actor.LN":"Khan"},{"TitleofFilm":1}).s ort({"YearofRelease":-1}).limit(2)

{ "\_id" : 21, "TitleofFilm" : "RaOne" }

{ "\_id" : 16, "TitleofFilm" : "Devdas" }

# List the titles of films acted by an actors „Shahrukh Khan‟ and „Madhuri Dixit‟

>

db.Film.find({$and:[{"Actor.FN":"Shahrukh","Actor.LN":"Khan"},{"Actor.FN":" Madhuri","Actor.LN":"Dixit"}]},{"TitleofFilm":1})

{ "\_id" : 15, "TitleofFilm" : "Dil To Pagal Hai" }

{ "\_id" : 16, "TitleofFilm" : "Devdas" }

# Some Extra Practice Queries

**Display a document of collection in formatted manner**

* db.Film.find({"\_id":17}).pretty()

# Display a specific field into Result by set value to 1

* db.collection.find({fieldname:value},{field:1})

# Ends with „s‟ and Stars with „T‟

* db.Film.find({"TitleofFilm":{$regex:"s$"}})
* db.Film.find({"TitleofFilm":{$regex:"/^T/"}})

# Sum of YearofRelease

* db.Film.aggregate([{$group:{\_id:null,total:{$sum:"$YearofRelease"}}}])

# Increament the YearofRelease

>

db.Film.aggregate([{$project:{TitleofFilm:1,total:{$add:["$YearofRelease",2 000]}}}])

# Find Count

* db.Film.find().count()

# Remove a document

* db.collection.remove({field:value})

# Update a document

* db.Film.update({\_id:21},{$set:{"Genre":["Action","SciFi"]}})

# Sort documents in ascending or descending order

* db.Film.find().sort({TitleofFilm:1})
* db.Film.find().sort({YearofRelease:-1})

db.collection1.aggregate([

{

$lookup:

{

from: "collection2", localField: "c1field", foreignField: "c2field", as: "formed collection name"

}

}

])

***Ahemadnagar Jilha Maratha Prasarak Samaj’s***

## Rajarshi Shahu Mahavidyalaya, Deolali Pravara

**Class: M.Sc(Computer Science)-I Assignment no: 2**

**Name: Sign:**

**Title: Company Database Date:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

C:\Users\user>cd C:\Program Files\MongoDB\Server\4.2\bin C:\Program Files\MongoDB\Server\4.2\bin>mongo

* show dbs
* use Company

switched to db Company

# Assignment 2

1. **Create a database with name „Company‟.**

# An „Employee‟ is a collection of documents with the following fields:

* 1. **Employee ID**
  2. **First Name**
  3. **Last Name**
  4. **Email**
  5. **Phone No.**
  6. **Address(House No,Street, City, State, Country, Pin-code)**
  7. **Salary**
  8. **Designation**
  9. **Experience**
  10. **Date of Joining**
  11. **Birthdate**

# A „Transaction‟ is a collection of documents with the following fields:

* 1. **Transaction Id,**
  2. **Transaction Date**
  3. **Name (First Name of employee who processed the transaction)**
  4. **Transaction Details (Item Id, Item Name, Quantity, Price)**
  5. **Payment (Type of Payment (Debit/Credit/Cash), Total amount paid, Payment Successful)**
  6. **Remark (Remark field can be empty.)**

# Queries:

1. **Insert at least 5 documents in „Employee‟ collection.**

>

[db.Employee.insert({\_id:1,FN:"Suresh",LN:"Wadkar",Email:"suresh123@gmail.co](mailto:suresh123@gmail.co) m",Phone:9876543210,Address:{HouseNo:2234,Street:56,City:"Mumbai",State:"Ma harashtra",Country:"India",Pincode:457213},Salary:15000,Designation:"Employ ee",Experience:2,DateofJion:new Date("2018-05-23"),BirthDate:new

Date("1995-03-12")})

WriteResult({ "nInserted" : 1 })

>

[db.Employee.insert({\_id:2,FN:"Mukesh",LN:"Ambani",Email:"mukesh123@gmail.co](mailto:mukesh123@gmail.co) m",Phone:9876543210,Address:{HouseNo:2864,Street:92,City:"Mumbai",State:"Ma harashtra",Country:"India",Pincode:451443},Salary:10000,Designation:"Manage r",Experience:20,DateofJion:new Date("1998-05-23"),BirthDate:new

Date("1975-03-12")})

WriteResult({ "nInserted" : 1 })

>

[db.Employee.insert({\_id:3,FN:"King",LN:"Kong",Email:"King123@gmail.com",Pho](mailto:King123@gmail.com) ne:9876543210,Address:{HouseNo:2523,Street:25,City:"Mumbai",State:"Maharash tra",Country:"India",Pincode:451443},Salary:8000,Designation:"Employee",Exp erience:1,DateofJion:new Date("2019-05-23"),BirthDate:new Date("1995-03- 12")})

WriteResult({ "nInserted" : 1 })

>

[db.Employee.insert({\_id:4,FN:"Mark",LN:"Zukerburg",Email:"mark123@gmail.com](mailto:mark123@gmail.com) ",Phone:9876543210,Address:{HouseNo:2523,Street:25,City:"Mumbai",State:"Mah arashtra",Country:"India",Pincode:451443},Salary:18000,Designation:"Supervi ser",Experience:4,DateofJion:new Date("2015-05-23"),BirthDate:new Date("1999-03-12")})

WriteResult({ "nInserted" : 1 })

>

[db.Employee.insert({\_id:5,FN:"Indiana",LN:"Johnes",Email:"johnes123@gmail.c](mailto:johnes123@gmail.c) om",Phone:9876543210,Address:{HouseNo:2523,Street:25,City:"Mumbai",State:"M aharashtra",Country:"India",Pincode:451443},Salary:1000,Designation:"FloorS uperviser",Experience:6,DateofJion:new Date("2013-05-23"),BirthDate:new Date("1999-03-12")})

WriteResult({ "nInserted" : 1 })

# Insert multiple documents (at least 10) into the

**„Transaction‟ collection by passing an array of documents to the db.collection.insert () method.**

* db.Transaction.insert([{\_id:201,TransDate:new Date("2017-09- 23"),Name:"Mukesh",TransDetails:{ItemId:11,ItemName:"Burger",Quantity:1000, Price:50},Payment:{TypeofPay:"Cash",TotalAmtPaid:40},Remark:"Pending"},{\_id

:202,TransDate:new Date("2018-03- 12"),Name:"Indiana",TransDetails:{ItemId:12,ItemName:"Pizza",Quantity:300,P

rice:299},Payment:{TypeofPay:"Credit",TotalAmtPaid:299},Remark:"Ok"},{\_id:2 03,TransDate:new Date("2019-08- 25"),Name:"Suresh",TransDetails:{ItemId:13,ItemName:"Vine",Quantity:100,Pri ce:700},Payment:{TypeofPay:"Debit",TotalAmtPaid:700},Remark:"Ok"},{\_id:204, TransDate:new Date("2016-09-29"),Name:"King",TransDetails:{ItemI d:14,ItemName:"Noodles",Quantity:50,Price:100},Payment:{TypeofPay:"Credit", TotalAmtPaid:90},Remark:"Pending"},{\_id:205,TransDate:new Date("2019-10- 12"),Name:"Mark",TransDetails:{ItemId:15,ItemName:"Coffee",Quantity:200,Pri ce:150},Payment:{TypeofPay:"Cash",TotalAmtPaid:150},Remark:"Ok"},{\_id:206,T ransDate:new Date("2000-06- 20"),Name:"King",TransDetails:{ItemId:16,ItemName:"Laptop",Quantity:50,Pric e:20000},Payment:{TypeofPay:"Debit",TotalAmtPaid:10000},Remark:"Pending"},{

\_id:207,TransDate:new Date("2005-12- 30"),Name:"Indiana",TransDetails:{ItemId:17,ItemName:"Diamond",Quantity:10, Price:99},Payment:{TypeofPay:"Cash",TotalAmtPaid:99},Remark:"Ok"},{\_id:208, TransDate:new Date("2013-09- 25"),Name:"Mark",TransDetails:{ItemId:18,ItemName:"BMWBike",Quantity:10,Pri ce:50},Payment:{TypeofPay:"Cash",TotalAmtPaid:50},Remark:"Ok"},{\_id:209,Tra nsDate:new Date("2008-02- 26"),Name:"Suresh",TransDetails:{ItemId:19,ItemName:"Shoe",Quantity:2000,Pr ice:1999},Payment:{TypeofPay:"Cash",TotalAmtPaid:1500},Remark:"Pending"},{\_ id:210,TransDate:new Date("2019-12- 28"),Name:"King",TransDetails:{ItemId:20,ItemName:"Book",Quantity:2000,P rice:200},Payment:{TypeofPay:"Debit",TotalAmtPaid:200},Remark:"Ok"}])

BulkWriteResult({

"writeErrors" : [ ], "writeConcernErrors" : [ ], "nInserted" : 10,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

# Display all the documents of both the collections in a formatted manner.

* db.Employee.find().pretty()

{

"\_id" : 1,

"FN" : "Suresh",

"LN" : "Wadkar",

"Email" : ["suresh123@gmail.com",](mailto:suresh123@gmail.com) "Phone" : 9876543210,

"Address" : {

"HouseNo" : 2234,

"Street" : 56, "City" : "Mumbai",

"State" : "Maharashtra", "Country" : "India", "Pincode" : 457213

},

"Salary" : 15000, "Designation" : "Employee", "Experience" : 2,

"DateofJion" : ISODate("2018-05-23T00:00:00Z"), "BirthDate" : ISODate("1995-03-12T00:00:00Z")

}

{

"\_id" : 2,

"FN" : "Mukesh",

"LN" : "Ambani",

"Email" : ["mukesh123@gmail.com",](mailto:mukesh123@gmail.com) "Phone" : 9876543210,

"Address" : {

"HouseNo" : 2864,

"Street" : 92, "City" : "Mumbai",

"State" : "Maharashtra", "Country" : "India", "Pincode" : 451443

},

"Salary" : 10000, "Designation" : "Manager", "Experience" : 20,

"DateofJion" : ISODate("1998-05-23T00:00:00Z"), "BirthDate" : ISODate("1975-03-12T00:00:00Z")

}

{

"\_id" : 3, "FN" : "King",

"LN" : "Kong",

"Email" : ["King123@gmail.com",](mailto:King123@gmail.com) "Phone" : 9876543210,

"Address" : {

"HouseNo" : 2523,

"Street" : 25, "City" : "Mumbai",

"State" : "Maharashtra", "Country" : "India", "Pincode" : 451443

},

"Salary" : 8000, "Designation" : "Employee", "Experience" : 1,

"DateofJion" : ISODate("2019-05-23T00:00:00Z"), "BirthDate" : ISODate("1995-03-12T00:00:00Z")

}

{

"\_id" : 4, "FN" : "Mark",

"LN" : "Zukerburg",

"Email" : ["mark123@gmail.com",](mailto:mark123@gmail.com) "Phone" : 9876543210,

"Address" : {

"HouseNo" : 2523,

"Street" : 25, "City" : "Mumbai",

"State" : "Maharashtra", "Country" : "India", "Pincode" : 451443

},

"Salary" : 18000, "Designation" : "Superviser", "Experience" : 4,

"DateofJion" : ISODate("2015-05-23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z")

}

{

"\_id" : 5,

"FN" : "Indiana",

"LN" : "Johnes",

"Email" : ["johnes123@gmail.com",](mailto:johnes123@gmail.com) "Phone" : 9876543210,

"Address" : {

"HouseNo" : 2523,

"Street" : 25, "City" : "Mumbai",

"State" : "Maharashtra", "Country" : "India", "Pincode" : 451443

},

"Salary" : 1000,

"Designation" : "FloorSuperviser", "Experience" : 6,

"DateofJion" : ISODate("2013-05-23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z")

}

* db.Transaction.find().pretty()

{

"\_id" : 201,

"TransDate" : ISODate("2017-09-23T00:00:00Z"),

"Name" : "Mukesh", "TransDetails" : {

"ItemId" : 11, "ItemName" : "Burger", "Quantity" : 1000,

"Price" : 50

},

"Payment" : {

"TypeofPay" : "Cash", "TotalAmtPaid" : 40

},

"Remark" : "Pending"

}

{

"\_id" : 202,

"TransDate" : ISODate("2018-03-12T00:00:00Z"),

"Name" : "Indiana", "TransDetails" : {

"ItemId" : 12, "ItemName" : "Pizza", "Quantity" : 300,

"Price" : 299

},

"Payment" : {

"TypeofPay" : "Credit", "TotalAmtPaid" : 299

},

"Remark" : "Ok"

}

{

"\_id" : 203,

"TransDate" : ISODate("2019-08-25T00:00:00Z"),

"Name" : "Suresh", "TransDetails" : {

"ItemId" : 13, "ItemName" : "Vine", "Quantity" : 100,

"Price" : 700

},

"Payment" : {

"TypeofPay" : "Debit", "TotalAmtPaid" : 700

},

"Remark" : "Ok"

}

{

"\_id" : 204,

"TransDate" : ISODate("2016-09-29T00:00:00Z"),

"Name" : "King", "TransDetails" : {

"ItemId" : 14, "ItemName" : "Noodles", "Quantity" : 50,

"Price" : 100

},

"Payment" : {

"TypeofPay" : "Credit", "TotalAmtPaid" : 90

},

"Remark" : "Pending"

}

{

"\_id" : 205,

"TransDate" : ISODate("2019-10-12T00:00:00Z"),

"Name" : "Mark", "TransDetails" : {

"ItemId" : 15, "ItemName" : "Coffee", "Quantity" : 200,

"Price" : 150

},

"Payment" : {

"TypeofPay" : "Cash", "TotalAmtPaid" : 150

},

"Remark" : "Ok"

}

{

"\_id" : 206,

"TransDate" : ISODate("2000-06-20T00:00:00Z"),

"Name" : "King", "TransDetails" : {

"ItemId" : 16, "ItemName" : "Laptop", "Quantity" : 50,

"Price" : 20000

},

"Payment" : {

"TypeofPay" : "Debit", "TotalAmtPaid" : 10000

},

"Remark" : "Pending"

}

{

"\_id" : 207,

"TransDate" : ISODate("2005-12-30T00:00:00Z"),

"Name" : "Indiana", "TransDetails" : {

"ItemId" : 17, "ItemName" : "Diamond", "Quantity" : 10,

"Price" : 99

},

"Payment" : {

"TypeofPay" : "Cash", "TotalAmtPaid" : 99

},

"Remark" : "Ok"

}

{

"\_id" : 208,

"TransDate" : ISODate("2013-09-25T00:00:00Z"),

"Name" : "Mark", "TransDetails" : {

"ItemId" : 18, "ItemName" : "BMWBike", "Quantity" : 10,

"Price" : 50

},

"Payment" : {

"TypeofPay" : "Cash", "TotalAmtPaid" : 50

},

"Remark" : "Ok"

}

{

"\_id" : 209,

"TransDate" : ISODate("2008-02-26T00:00:00Z"),

"Name" : "Suresh", "TransDetails" : {

"ItemId" : 19, "ItemName" : "Shoe", "Quantity" : 2000,

"Price" : 1999

},

"Payment" : {

"TypeofPay" : "Cash", "TotalAmtPaid" : 1500

},

"Remark" : "Pending"

}

{

"\_id" : 210,

"TransDate" : ISODate("2019-12-28T00:00:00Z"),

"Name" : "King", "TransDetails" : {

"ItemId" : 20, "ItemName" : "Book", "Quantity" : 2000,

"Price" : 200

},

"Payment" : {

"TypeofPay" : "Debit", "TotalAmtPaid" : 200

},

"Remark" : "Ok"

}

# Update salary of all employees by giving an increment of Rs. 4000.

>

db.Employee.aggregate([{$project:{"Name":1,Salary:{$add:["$Salary",4000]}}}

])

{ "\_id" : 1, "Salary" : 19000 }

{ "\_id" : 2, "Salary" : 14000 }

{ "\_id" : 3, "Salary" : 12000 }

{ "\_id" : 4, "Salary" : 22000 }

{ "\_id" : 5, "Salary" : 5000 }

# Update the remark for transaction id 201.

* db.Transaction.update({\_id:201},{$set:{"Remark":"Ok"}}) WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Update designation of an employee named “Mark” from supervisor to manager.

* db.Employee.update({"FN":"Mark"},{$set:{"Designation":"Manager"}}) WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Update designation of an employee having Employee Id as 4.

* db.Employee.update({"\_id":4},{$set:{"Designation":"Superviser"}}) WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Change the address of an employee having Employee Id as 4.

>

db.Employee.update({"\_id":4},{$set:{Address:{HouseNo:9843,Street:45,City:"P une",State:"Maharashtra",Country:"India",Pincode:445333}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

# Delete transaction made by “ ” employee on the given date.

* db.Transaction.remove({$and:[{"Name":"Suresh"},{"TransDate":new Date("2008-02-26")}]})

WriteResult({ "nRemoved" : 1 })

# Delete all the employees whose first name starts with „K‟.

* db.Employee.remove({FN:{$regex:/^K/}}) WriteResult({ "nRemoved" : 1 })

***Ahemadnagar Jilha Maratha Prasarak Samaj’s***

## Rajarshi Shahu Mahavidyalaya, Deolali Pravara

**Class: M.Sc(Computer Science)-I Assignment no: 4**

**Name: Sign:**

**Title: Company Database Date:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

C:\Users\user>cd C:\Program Files\MongoDB\Server\4.2\bin C:\Program Files\MongoDB\Server\4.2\bin>mongo

* show dbs
* use Company

switched to db Company

# This assignment is based on „Company‟ database having collections „Employee‟ and „Transaction‟.Using MongoDB Aggregate framework

1. **Find employees having designation as either „manager‟ or**

# „floor supervisor‟.

>

db.Employee.find({$or:[{"Designation":"Manager"},{"Designation":"FloorSuper viser"}]})

{ "\_id" : 2, "FN" : "Mukesh", "LN" : "Ambani", "Email" : ["mukesh123@gmail.com",](mailto:mukesh123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2864, "Street" : 92, "City" : "Mumbai", "State" : "Maharashtra", "Country"

: "India", "Pincode" : 451443 }, "Salary" : 10000, "Designation" : "Manager", "Experience" : 20, "DateofJion" : ISODate("1998-05- 23T00:00:00Z"), "BirthDate" : ISODate("1975-03-12T00:00:00Z") }

{ "\_id" : 5, "FN" : "Indiana", "LN" : "Johnes", "Email" : ["johnes123@gmail.com",](mailto:johnes123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2523, "Street" : 25, "City" : "Mumbai", "State" : "Maharashtra", "Country"

: "India", "Pincode" : 451443 }, "Salary" : 1000, "Designation" : "FloorSuperviser", "Experience" : 6, "DateofJion" : ISODate("2013-05- 23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z") }

# Find an employee whose name ends with "k" and print the output in json format.

* db.Employee.find({"FN":{$regex:"k$"}})

{ "\_id" : 4, "FN" : "Mark", "LN" : "Zukerburg", "Email" : ["mark123@gmail.com",](mailto:mark123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2523, "Street" : 25, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 451443 }, "Salary" : 18000, "Designation" : "Superviser", "Experience" : 4, "DateofJion" : ISODate("2015-05- 23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z") }

# Display the name of an employee whose salary is greater than 15000 using a MongoDB cursor.

* db.Employee.find({"Salary":{$gt:15000}},{"FN":1})

{ "\_id" : 4, "FN" : "Mark" }

# Sort the employees in the descending order of their designation.

* db.Employee.find({}).sort({"Designation":-1})

{ "\_id" : 4, "FN" : "Mark", "LN" : "Zukerburg", "Email" : ["mark123@gmail.com",](mailto:mark123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2523, "Street" : 25, "City" : "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 451443 }, "Salary" : 18000, "Designation" : "Superviser", "Experience" : 4, "DateofJion" : ISODate("2015-05- 23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z") }

{ "\_id" : 2, "FN" : "Mukesh", "LN" : "Ambani", "Email" : ["mukesh123@gmail.com",](mailto:mukesh123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2864, "Street" : 92, "City" : "Mumbai", "State" : "Maharashtra", "Country"

: "India", "Pincode" : 451443 }, "Salary" : 10000, "Designation" : "Manager", "Experience" : 20, "DateofJion" : ISODate("1998-05- 23T00:00:00Z"), "BirthDate" : ISODate("1975-03-12T00:00:00Z") }

{ "\_id" : 5, "FN" : "Indiana", "LN" : "Johnes", "Email" : ["johnes123@gmail.com",](mailto:johnes123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2523, "Street" : 25, "City" : "Mumbai", "State" : "Maharashtra", "Country"

: "India", "Pincode" : 451443 }, "Salary" : 1000, "Designation" : "FloorSuperviser", "Experience" : 6, "DateofJion" : ISODate("2013-05- 23T00:00:00Z"), "BirthDate" : ISODate("1999-03-12T00:00:00Z") }

{ "\_id" : 1, "FN" : "Suresh", "LN" : "Wadkar", "Email" : ["suresh123@gmail.com",](mailto:suresh123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2234, "Street" : 56, "City" : "Mumbai", "State" : "Maharashtra", "Country"

: "India", "Pincode" : 457213 }, "Salary" : 15000, "Designation" : "Employee", "Experience" : 2, "DateofJion" : ISODate("2018-05- 23T00:00:00Z"), "BirthDate" : ISODate("1995-03-12T00:00:00Z") }

{ "\_id" : 3, "FN" : "King", "LN" : "Kong", "Email" : ["King123@gmail.com",](mailto:King123@gmail.com) "Phone" : 9876543210, "Address" : { "HouseNo" : 2523, "Street" : 25, "City"

: "Mumbai", "State" : "Maharashtra", "Country" : "India", "Pincode" : 451443 }, "Salary" : 8000, "Designation" : "Employee", "Experience" : 1, "DateofJion" : ISODate("2019-05-23T00:00:00Z"), "BirthDate" : ISODate("1995-03-12T00:00:00Z") }

# Count the total number of employees in a collection.

* db.Employee.find({}).count() 5

# Calculate the sum of total amount paid for all the transaction documents.

>

db.Transaction.aggregate([{$group:{\_id:null,total:{$sum:"$Payment.TotalAmtP aid"}}}])

{ "\_id" : null, "total" : 13128 }

# Calculate the sum of total amount paid for each payment type.

>

db.Transaction.aggregate([{$group:{\_id:"$Payment.TypeofPay",total:{$sum:"$P ayment.TotalAmtPaid"}}}])

{ "\_id" : "Credit", "total" : 389 }

{ "\_id" : "Debit", "total" : 10900 }

{ "\_id" : "Cash", "total" : 1839 }

# Find the transaction id of the latest transaction.

* db.Transaction.find({},{\_id:1}).sort({"TransDate":-1}).limit(1)

{ "\_id" : 210 }