Lab-12 Aggregate method

Mongo

DB

Perform the following queries using Aggregate:

Part – A Create collection named “Student” and insert following 10 documents into it.

1. Display distinct city.

db.Student.aggregate([

{$group : {\_id : '$CITY'}},

{$project : {\_id : 0, CITY : '$\_id'}}

])

2. Display city wise count of number of students.

db.Student.aggregate([

{$group : {\_id : '$CITY', count : {$sum : 1}}},

])

3. Display sum of fees in your collection.

db.Student.aggregate([

{$group : {\_id : null, Fees : {$sum : '$Fees'}}},

])

4. Display average of fees in your document.

db.Student.aggregate([

{$group : {\_id : null, Fees : {$avg : '$FEES'}}},

])

5. Display maximum and minimum fees of your document.

db.Student.aggregate([

{$group : {\_id : null, MaxFees : {$max : '$FEES'}, MinFees : {$min : '$FEES'}}},

])

6. Display city wise total fees in your collection.

db.Student.aggregate([

{$group : {\_id : '$CITY', TotalFees : {$sum : '$FEES'}}},

])

7. Display gender wise maximum fees in your collection.

db.Student.aggregate([

{$group : {\_id : '$GENDER', MaxFees : {$max : '$FEES'}}},

])

8. Display city wise maximum and minimum fees.

db.Student.aggregate([

{$group : {\_id : '$CITY', MaxFees : {$max : '$FEES'}, MinFees : {$min : '$FEES'}}},

])

9. Display count of persons lives in Baroda city in your collection.

db.Student.aggregate([

{$match : {CITY : 'Baroda'}},

{$group : {\_id : null, COUNT : {$sum : 1}}},

])

10. Display average fees of Rajkot city.

db.Student.aggregate([

{$match : {CITY : 'Rajkot'}},

{$group : {\_id : null, AVGFEES : {$avg : '$FEES'}}},

])

11. Count the number of male and female students in each Department

db.Student.aggregate([

{$group : {\_id :{Department :'$DEPARTMENT', Gender : '$GENDER'}, count : {$sum : 1}}},

])

12. Find the total Fees collected from each Department.

db.Student.aggregate([

{$group : {\_id :{Department :'$DEPARTMENT'}, TotalFeesPerDept : {$sum : '$FEES'}}},

])

13. Find the minimum Fees paid by male and female students in each City.

db.Student.aggregate([

{$group : {\_id :{City :'$CITY'}, MinFEES : {$min : '$FEES'}}},

])

14. Sort students by Fees in descending order and return the top 5.

db.Student.aggregate([

{$group : {\_id : '$ROLLNO', TotalFEES : {$sum : '$FEES'}}}, {$sort : {TotalFEES : -1 }},{$limit : 5},

])

15. Group students by City and calculate the average Fees for each city, only including cities with more than 1 student.

db.Student.aggregate([

{$group : {\_id : '$CITY',COUNT : {$sum : 1} ,AVGFEES : {$avg : '$FEES'}}},

{$match : {COUNT : {$gt : 1}}}

])

16. Filter students from CE or Mechanical department, then calculate the total Fees.

db.Student.aggregate([

{$match : {DEPARTMENT : {$in : ["CE", "Mechanical"] }}},

{$group : {\_id : null ,TotalFEES : {$sum : '$FEES'}}}

])

17. Count the number of male and female students in each Department.

db.Student.aggregate([

{$group : {\_id : { Department :'$DEPARTMENT', Dender : '$GENDER'},count : {$sum : 1}}}

])

18. Filter students from Rajkot, then group by Department and find the average Fees for each department.

db.Student.aggregate([

{$match : {CITY : {$in : ["Rajkot"] }}},

{$group : {\_id : '$DEPARTMENT' ,AVERAGEFEES : {$avg : '$FEES'}}}

])

19. Group by Sem and calculate both the total and average Fees, then sort by total fees in descending order.

db.Student.aggregate([

{$group : {\_id : '$SEM', TotalFEES : {$sum : '$FEES'}, AVERAGEFEES : {$avg : '$FEES'}}}, {$sort : {TotalFEES : -1 }}

])

20. Find the top 3 cities with the highest total Fees collected by summing up all students' fees in those cities'

db.Student.aggregate([

{ $group: { \_id: "$CITY", totalFees: { $sum: "$FEES" } } },

{ $sort: { totalFees: -1 } },

{ $limit: 3 }

]);

Part – B

1. Create a collection named” Stock.”

2. Insert below 9 documents into the “Stock” collection.

{ "\_id": 1,

"company": "Company-A",

"sector": "Technology",

"eps": 5.2,

"pe": 15.3,

"roe": 12.8,

"sales": 300000,

"profit": 25000

}

{ "\_id": 2,

"company": "Company-B",

"sector": "Finance",

"eps": 7.1,

"pe": 12.4,

"roe": 10.9,

"sales": 500000,

"profit": 55000

}

{ "\_id": 3,

"company": "Company-C",

"sector": "Retail",

"eps": 3.8,

"pe": 22.1,

"roe": 9.5,

"sales": 200000,

"profit": 15000

}

{ "\_id": 4,

"company": "Company-D",

"sector": "Technology",

"eps": 5.2,

"pe": 15.3,

"roe": 12.8,

"sales": 300000,

"profit": 25000

}

{ "\_id": 5,

"company": "Company-E",

"sector": "Finance",

"eps": 7.1,

"pe": 12.4,

"roe": 10.9,

"sales": 450000,

"profit": 40000

}

{ "\_id": 6,

"company": "Company-F",

"sector": "Healthcare",

"eps": 3.8,

"pe": 18.9,

"roe": 9.5,

"sales": 500000,

"profit": 35000

}

{ "\_id": 7,

"company": "Company-G",

"sector": "Retail",

"eps": 4.3,

"pe": 22.1,

"roe": 14.2,

"sales": 600000,

"profit": 45000

}

{

"\_id": 8,

"company": "Company-H",

"sector": "Energy",

"eps": 6.5,

"pe": 10.5,

"roe": 16.4,

"sales": 550000,

"profit": 50000

}

{

"\_id": 9,

"company": "Company-I",

"sector": "Consumer Goods",

"eps": 2.9,

"pe": 25.3,

"roe": 7.8,

"sales": 350000,

"profit": 20000

}

3. Calculate the total sales of all companies

4. Find the average profit for companies in each sector.

5. Get the count of companies in each sector/

6. Find the company with the highest PE ratio.

7. Filter companies with PE ratio greater than 20.(Use: Aggregate)

8. Calculate the total profit of companies with sales greater than 250,000.

9. Project only the company name and profit fields.(Use: Aggregate)

10. Find companies where EPS is greater than the average EPS.

11. Group companies by sector and get the maximum sales in each sector.

12. Calculate the total sales and total profit of companies in each sector.

13. Sort companies by profit in descending order.(Use: Aggregate)

14. Find the average ROE across all companies.

15. Group companies by sector and calculate both the minimum and maximum EPS.

Part – C

1. Count the number of companies with profit greater than 30,000.

2. Get the total profit by sector and sort by descending total profit.

3. Find the top 3 companies with the highest sales.

4. Calculate the average PE ratio of companies grouped by sector.

5. Get the sum of sales and profit for each company.

6. Find companies with sales less than 400,000 and sort them by sales.

7. Group companies by sector and find the total number of companies in each sector.

8. Get the average ROE for companies with sales greater than 200,000.

9. Find the maximum profit in each sector.

10. Get the total sales and count of companies in each sector.

11. Project fields where profit is more than 20,000 and only show company and profit.

12. Find companies with the lowest ROE and sort them in ascending order.(Use: Aggregate)